①

83 12 27 083

FIRST QUARTERLY PROGRESS REPORT ON THE

DARPA INTERNET PROJECT

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION U LIMITED

$MDA\ 90\ 3-83-C-\ 0034$

$A03674$

31 March 1983

David L. Mills
&
Zorica Avramovic

M/A - COM Linkabit
Eastern operations

Prepared for:

Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

DTIC

S D

DEC 2 7 1983

H

First Quarterly Progress Report on the DARPA Internet Project
David L. Mills and Zorica Avramovic
M/A-COM LINKABIT Eastern Operations
31 March 1983

Contents   Page

List of Figures   Page

1. INTRODUCTION

This first Quarterly Project Report on LINKABIT's contribution to the Defense Advanced Research Projects Agency (DARPA) Internet Program covers the period from 22 December 1982 through 21 March 1983. The work was carried out at LINKABIT's Eastern Operations facility in McLean, Virginia. Contributing to the effort were David L. Mills, Project Engineer and Manager, and Zorica Avramovic, Senior Engineer. Robert Enger and Walter Roehr provided technical support and assistance, and Doreen Klein provided secretarial support.

LINKABIT's support of the Internet Program is concentrated in the areas of protocol design, implementation, testing, and evaluation. In addition, LINKABIT staff are providing integration and support services for certain computer systems to be installed at DARPA sites in Washington, D.C., and Stuttgart, West Germany.

During the period covered by this report, LINKABIT organized the project activities and established staff responsibilities. Several computers and peripheral devices were made available from Government sources for use in protocol development and network testing. Considerable time was devoted to installing this equipment, integrating the software, and testing it with the Internet system.

The VOS/BOS Fuzzball system was selected as the principle software tool for the protocol development and network testing. This system was implemented first at the University of Maryland and further developed at COMSAT Laboratories. A number of enhancements have been added by LINKABIT personnel for the specific needs of protocol development and network testing and, in addition, for operational use at DARPA sites. These enhancements are described later in this document.

An important part of LINKABIT's contribution to the Internet Program is support for occasional demonstrations and tests and attendance at regular meetings of the Internet Control and Configuration Board (ICCB), SATNET Program, Multimedia Program and Research Group (RG). LINKABIT provided support for a demonstration of the Internet system at a symposium sponsored by SHAPE Technical Center (STC) in November 1982, and LINKABIT personnel attended the ICCB, SATNET and RG meetings at DFVLR (Munich) in September 1982--both at no cost to the Government. During this reporting period, LINKABIT personnel attended the ICCB, SATNET and RG meetings at LINKABIT Headquarters (San Diego) in January 1983 and the ICCB and SATNET meetings at MIT (Cambridge) in March 1983.

Section 2 of this report summarizes specific items of progress, including the status of equipment integration, software enhancements, and testing activities. Section 3 briefly presents plans for the next quarter of the project, and Section 4 summarizes the DCNET architecture and implementation. The appendix provides a user reference guide for the system.

2. ACCOMPLISHMENTS

Accomplishments during the reporting period included the installation and test of a number of LSI-11 systems to be used as personal workstations, connected via the Distributed Computer Network (DCN) local network to the Internet system. In addition, software had to be adapted and configured for these machines. Finally, the system had to be tested with the Internet system and validated with other experimental systems and networks connected to the Internet system.

The principal issue to arise from the ICCB meetings was the early development and deployment of the Exterior Gateway Protocol (EGP), which is to be used to support gateways other than the Bolt Beranek and Newman (BBN) standard "core" gateways. An important activity during this period was directed toward the design of software modules to support EGP and coordination of the development with BBN and MIT personnel.

The network testing and evaluation effort focused on a set of experiments with the File Transfer Protocol (FTP) and on preparation for the STC demonstration. The results of these investigations provided insight into the problems often experienced in day-to-day operations over Internet system paths.

2.1. Equipment Integration

The Government-furnished equipment made available to LINKABIT included a number of LSI-11 computer systems, including video terminals, network interfaces, and special-purpose peripheral devices. Among the peripheral devices were a Dacom/Rapifax digital facsimile machine and a Lincoln Laboratories speech codec for use in multimedia system development. Communication equipment included 1200-bps dial-up modems and a set of error-control units (ECUs) for connection to an ARPANET IMP.

This equipment was assembled and integrated as a local network and connected to the ARPANET via the MITRE IMP in McLean, Virginia. The local network uses the DCN architecture, in which one of the LSI-11s is configured as a standard "core" gateway and another is configured as a program-development machine. In addition, two other LSI-11 systems were provided at no cost to the government and connected to the DCN for use in

general word-processing and network-testing applications.
Additional modems and video terminals have also been provided
from LINKABIT's capital equipment sources.

The local network has been designated DCNET and assigned the
address 128.4 in the Network Information Center (NIC) data base.
The LSI-11 hosts have been designated DCN1 (LINKABIT,
DCN-GATEWAY) for the ARPANET gateway, DCN3 for the
program-development machine, DCN5 for the word-processing
machine, and DCN6 (BACKROOM) for the network-testing machine.
All except DCN6 are interconnected by high-speed DMA interfaces,
while DCN6 is connected by a 1200-bps dial-up circuit and DCN1 is
connected to the MITRE IMP via a 9600-bps leased circuit.

One of the useful features of the DCN architecture used
heavily in prior experiments has been the precision with which
coordinated time can be maintained. This was ensured on a
continuing basis as a result of the installation of an NBS radio
clock. The clock contains a WWVB time-code receiver,
microprocessor control, and serial interface. The device is now
connected to the DCN1 host and used to synchronize--to within a
few milliseconds--all DCN hosts through the local-network
protocols.

2.2. Software Integration

The software integration effort included reconfiguration of
the Fuzzball software to operate with a new network number plus
certain changes and additions to the IP/TCP modules. In
addition, several new features were added to support multiple
users, new mail functions, and new services.

2.2.1. IP/TCP Modules

An IP reassembly function was added to the IP module. This
function operates to reassemble fragments into a complete IP
datagram in the following way: For TCP services, the fragments
are linked together and passed to the TCP module where they are
copied into the TCP reassembly buffer in user space for later
delivery to the client process. For datagram services (e.g.,
UDP) the fragments are linked together and passed to the client
process where they are copied into a user-space buffer as
required. With this approach the fragments are not reconstituted
in a linear buffer and do not require scarce buffer resources in
the operating-system kernel.

The IP option interface was rebuilt to provide for
user-specifiable options. A mechanism was provided for a user
process to specify and interpret options. At the TCP level, a
mechanism was provided to specify the TCP Max-Size option, which
can be used to limit the maximum TCP segment size on a

per-connection basis. This mechanism was found necessary for communicating with the BBN VAX TCP implementation, which, in the absence of this option can use datagrams larger than 576 octets-- the maximum size required by the IP specification.

2.2.2. New Services

New protocol servers for user-datagram protocol (UDP) time-server and name-server functions were added to the system. This required redesigning the user process interface and constructing the servers themselves. The time server is compatible with IEN-142 and provides time as a 32-bit quantity in seconds past midnight, 1 January 1900. It retains the accuracy of the DCN time-synchronization protocols, but has a resolution of one second. The name server is compatible with IEN-116 and provides the address of a host given as an argument in the request datagram. Both servers are supported by a single dedicated user process, which must be configured in each host where the service is required. Presently, DCN1 and DCN6 are so configured.

The new servers were tested with BBN (HP-3000), MIT (Multics), ISI (TOPS-20), and Purdue (VAX) implementations, as well as other Fuzzballs in the United States and Norway. Several bugs were found and fixed in various servers; however, some incompatibilities still exist: For instance, MIT name servers operate using a non-standard port. It also was found that times provided by various hosts are often grossly inaccurate, ranging up to five-minute descrepancies in some cases. A sidelight of these experiments was a calibration of the drift of the London power-grid frequency, as measured via a Fuzzball at University College London.

As part of the name-server development, considerable interaction was had with Network Information Center (NIC) personnel who are developing the NIC version of the name server. Assistance was extended by LINKABIT personnel on several occasions to provide host name/address information, to review and correct the data-base information, and to test the NIC (Foonly) server. Name servers and user programs (dubbed "name callers") were also tested with BBN (HP-3000) and MIT (Multics) hosts.

2.2.3. System Support

The most significant development was the completion of virtual-memory features for multiple-user support. These features provide for individually relocated user processes, each of which shares a common, re-entrant operating-system emulator. The emulator supports all features necessary to run the DEC RT-11 operating-system utilities, editors, compilers, and user programs.

Each of these user processes can run separate copies of the
TELNET, FTP, and mail programs as well. Full 22-bit addressing
is supported.

A new interprocess communication (IPC) system was installed
to replace the bulky and slow existing system. The new system
provides a two-fold increase in speed for asynchronous terminals
and lines. In general, Fuzzballs with the new IPC system can
sustain speeds of between 2400 bps (LSI-11/2) and 4800 bps
(LSI-11/23) on a program-interrupt basis for any of several
synchronous and asynchronous communication devices.

A driver for the Lincoln Laboratories Linear Predictive Codec
Microprocessor (LPCM) was constructed for the DEC DPV11 serial
synchronous line interface. This effort was the result of
reliability problems with the Lincoln Laboratories interface and
power supply formerly used for this purpose. The driver operates
with file formats established by Carnegie Mellon in real time at
speeds up to 4800 bps.

Support for the Peritek bit-map color display was rebuilt and
new features were added. This device provides a 512 x 640 pel
display with three bits per pel and a palette of 4096 colors. A
common supporting module was constructed that provides
vector-graphic emulation for the Tektronix 4000-series terminals,
bit-map displays compatible with exisitng multimedia editors, and
ordinary text in three fonts. This module was incorporated into
the TELNET user program as an auxilliary output device and into a
special graphics server for use with the ISI Briefing Aids system.

The TELNET user and server programs were considerably
enhanced. Features to transmit and receive files via the TELNET
connection were added to the TELNET user program. In addition
features were added to the TELNET server to support several
standard functions such as echo, test, discard, time, and
others. The operator-intercom facility, often found useful in
test coordination, was enhanced to provide transparent loading of
speech interface units and other programmable devices via the
TELNET connection. Full flow-control features were added to the
serial-line drivers so that the Fuzzballs could be used as PADs
to access other terminal and hosts via the TELNET connection.
Finally, support for TELNET option negotiations was added to both
user and server programs.

2.2.4. Mail System Support

Development of the electronic-mail system continued, with the
conversion to SMTP (RFC-821) and new header formats (RFC-822).

The older MTP has now been phased out. Domain-style names are used in SMTP and are optional in the user-interface software. Many bugs were found and fixed, both in the Fuzzball software and in other systems.

The major new features of the mail system are the capabilities to answer and forward mail. These features operate similarly to the TOPS-20 MSG program. The "answer" feature involves parsing the header of a delivered message for recipient mailboxes. If the "Reply-to:" field does not exist, the address in the mandatory "From:" field is used. Other recipients are obtained by parsing the "To" and "cc:" fields depending on the specified option. The subject title is preceded by "Re:" to alert the recipients that this is a reply message. The user need only complete the body of the reply, after which the normal SMTP processing causes it to be sent.

The "forward" feature involves collecting a sequence of messages as specified by the user, appending comments, and affixing a header as determined by the user. The header is created as for any other regular message that needs to be sent. The body of the message is then entered by the user and is automatically followed by the forwarded message(s).

2.3. Support for DARPA Installations

As part of a planned installation in Washington, D.C., and Stuttgart, West Germany, a level of effort has been maintained to integrate these systems with the Fuzzball software. In addition, Fuzzball configurations were prepared (at no cost to the Government) for the STC demonstration and for several other organizations that requested them and received DARPA approval.

The STC demonstration involved a Fuzzball installed temporarily at The Hague, Netherlands, connected via leased line to a gateway at Royal Signals and Radar Establishment (RSRE) in Malvern, UK. The gateway provided connection to the UCL Gateway in London, then via SATNET to ARPANET service hosts. DCN local-net protocols were used above X.25 link-level (LAPB) protocols on the STC - RSRE path, with other established protocols being used elsewhere.

A device driver for the RSRE X.25 interface was constructed in support of this demonstration. The driver was integrated into a software package and sent for testing at BBN, who tested it using a system that will be installed in Stuttgart. The system was sent to The Hague for use in the demonstration and was subsequently shipped to Stuttgart. LINKABIT personnel assisted in testing and debugging, using the Internet system for remote debugging and directly by telephone.

The Stuttgart configuration supports six users, in addition to a multipurpose TCP server providing TELNET, FTP, and mail services. A configuration was completed and tested on DCN5. Subsequently, floppy disks were prepared and sent via DFVLR to Stuttgart for installation. Since special interfaces to connect the Stuttgart machine to the DCN clone now operating at DFVLR (Oberpfaffenhofen) are not yet available, a driver for the standard DEC DPV11 serial synchronous line interface was constructed and tested with other DCN clones connected via the DCN.GATEWAY DPV11s were then purchased for use on the Stuttgart - Oberpfaffenhofen link, and the drivers were integrated with the help of DFVLR personnel.

In response to the local need for a high-speed DMA serial synchronous connection for DCN hosts, as well as the need for a backup for the relatively expensive Associated Computer Consultants XQ/CP interfaces planned for use on the Stuttgart - Oberpfaffenhofen link, a driver for the new DEC DMV11 interface was constructed. The DMV11 utilizes an on-board microprocessor and ROM to implement the DDCMP link-level protocol used with their Digital Network Architecture (DNA) products and operates to 56 Kbps over either half-duplex or full-duplex links. The driver was tested with the DCN clone at Ford Moter Company and was subsequently put into full-time operation.

2.4. Gateway Protocols, Design, and Implementation

One of the main goals of the LINKABIT contribution to the Internet Program is to improve the understanding of the issues involved in deploying large, heirarchically structured networks. In the case of the Internet system, these issues come to focus in the design and implementation of the Internet gateway. Past development of the DCN architecture has resulted in a prototype gateway that supports all standard "core" gateway functions except the Gateway Monitoring Protocol (GMP). However, the standard Gateway Gateway Protocol (GGP) used to convey routing and connectivity information between the core gateways has been found inadequate for use in large, proliferated systems. The DCN prototype gateway has been used extensively as a tool to study the problems, instrument the performance and test the functionality of proposed improvements.

During this reporting period the DCN experimental gateway was modified to improve performance and to support new features required for the implementation of newly proposed gateway protocols such as the EGP. In particular, the network-routing function was redesigned, with implementation scheduled for the next quarter. The kernel-resident code has been moved to a dedicated user process for ease of debugging and to free needed kernel address space for additior buffers. Finally, the

gateway support module was rebuilt to operate with multiple routing algorithms and protocols running at the same time, with provisions for coupling routing information from one set of algorithms and protocols to the others in a controlled manner.

A prototype implementation of the new gateway support module was constructed and installed on the DCN6 network-testing host for evaluation. It now runs GGP with DCN-GATEWAY, but does not exchange significant routing information. Development will continue during the next quarter. Section 3 contains an outline of the architecture and functionality of the proposed model.

2.5. Experiments

Another important aspect of LINKABIT's participation in the Internet Program is a continuing series of network-testing experiments designed to validate protocol functionality, detect and resolve problems and assess network performance. These experiments have been useful in the development of new protocols and have built (or destroyed) confidence in intricate distributed protocol architectures, where a certain degree of experimentation has been necessary to establish conceptual feasibility and to refine the operational model.

During this reporting period, several experiments were conducted using the File Transfer Protocol (FTP) defined in RFC-765 and the DCN implementation, with peers represented by the BBN TOPS-20 implementation and the BBN VAX implementation. These tests were specifically designed to detect instances of protocol violations and to assess the performance of bulk-transport service via gateway paths.

The results in this initial suite of tests have been disappointing. The TOPS-20 implementation was found to be grossly defective in several areas, notably the connection-close sequence, which is vital for reliable data transfer in some modes, and the support for multiple-get/send sequences. In addition, the TCP connection parameters, such as initial retransmission-timeout, packetization, and retransmission strategy are unworkable on paths including low-speed tandem links such as found in some DCN links and in SATNET. Throughput experiments on paths between the NDRE (Norway) Fuzzball and the ISID TOPS-20 averaged about 1000 bps during the experiments, well below the 5000-6000 bps capability of that path. On the other hand, throughputs between the Purdue BBN VAX and the DCN1 fuzzball averaged about 5400 bps, which is near the maximum attainable with the 9600-bps ARPANET access line.

These experiments will be continued while efforts to fix the various implementations continue. It is expected that the TOPS-20 problems can be corrected by extending the retransmission timeouts and changing the parameters of the timeout-estimation algorithm.

Another experiment designed to test the capability of gateways connected by relatively slow access lines (9600 bps), such as the case with DCN-GATEWAY and NDRE-GATEWAY, revealed significant problems when routing-update "spasms" occur as all gateways in the system attempt to converge their routing tables in the face of loss of connectivity when a gateway fails. It was calculated and later confirmed by observation that the speed with which the gateway system could respond to sudden loss of connectivity was limited by the access lines.

In the case of the existing configuration, for example, the GGP messages required to converge the routing information required over 20 seconds of line time. Efforts are under way to alleviate this problem in two areas: Plans are being made to upgrade the DCN-GATEWAY access line to 56 Kbps and to convert DCN-GATEWAY to EGP, which does not suffer from this problem.

Finally, an ongoing experiment was continued with personnel at the University of Maryland in the development of IP/TCP support for the Univac 1100-series mainframes. This development, supported by the University at no cost to the Government, resulted in the development of IP/TCP capability as part of the EXEC-9 operating system for that machine. It was tested during this quarter using University-owned Fuzzballs and then with the LINKABIT DCN Fuzzballs. It is now in testing at the request of the U.S. Navy with other Internet hosts. In addition, collaboration with Ford Aerospace continued in the testing of IP/TCP support for the 3-COM Unix system for the VAX and with the Ford Motor Company in the resolution of issues involved in connecting DCN subnets to the system.

3. PLANS FOR THE NEXT QUARTER

The most important goals for the next quarter of this project to implement a EGP gateway to test the functionality of the EGP specification RFC-827 and to work out the details of the protocol itself. Experiments are to continue with FTP in order to assess the effectiveness of TOPS-20 changes and to improve throughput, especially on SATNET paths. Finally, efforts will continue to upgrade the Fuzzball system and to integrate new features required by the DARPA installations.

4. DCNET ARCHITECTURE AND IMPLEMENTATION

The DCNET Fuzzball system consists of a network architecture together with a software implementation developed over the last several years by LINKABIT personnel under DARPA sponsorship. It has been used extensively for testing, evaluation, and experimentation with other sytems in the defense and industrial communities conforming to the DARPA Internet protocols. It supports virtual-terminal, file-transfer, electronic-mail and general-purpose program-development functions in PDP11 and LSI-11 computers with varying configurations and applications. The software implementation consists of a general-purpose operating system that emulates the DEC standard RT-11 operating system, together with a package of software modules operating at various levels relative to the ISO model. The system can be configured for several distinct functions, including terminal-access controller, internet gateway, general-purpose host, and programmer's workbench.

The Distributed Computer Network (DCNET) is a generic label for any of several network clones based on a technology developed first at the University of Maryland and continuing in recent years under DARPA's Internet Program. The primary application of the DCNET is as a vehicle for experimentation and evaluation of novel protocols and network architectures--in particular, the various protocols developed by the Internet Program.

A DCNET clone consists of a number of PDP11-compatible hosts with both general-purpose peripherals and special devices including bit-map graphics displays, digital facsimile machines and speech codecs. The hosts, sometimes called "fuzzballs," are generally used as single-user personal computers, although multiple-user configurations can be readily supported with the requisite hardware. A typical configuration includes an LSI-11/2 or LSI-11/23 with 30K to 124K words of memory, dual floppy or Winchester disk drives, network interface device, and operator terminal.

DCNET hosts can be connected by a variety of communication links and interface devices, including simple point-to-point synchronous and asynchronous serial lines, 16-bit parallel interfaces, and ARPANET 1822 interfaces using both program-interrupt and direct-memory-access hardware. Some configurations utilize a contention-bus cable system for this purpose. Automatic routing, configuration-control, and time-synchronization features are common to all configurations.

Various DCNET clones have been established in several configurations and connected either directly to an ARPANET IMP or SRI Port Expander or indirectly via a MACRO-11 Internet Gateway to the internet system. In addition, a DCNET host can be

configured to act as an internet terminal-access controller,
port expander and/or gateway in addition to performing normal
host functions.

The following subsections generally describe the DCNET
Fuzzball system and include an overview of the system
architecture and capabilities. Included also are references to
other documents that describe the system in more detail, as
well as short descriptions of the user-level protocol modules
currently implemented.

4.1. IP/TCP Support Software

The software system consists of a package of MACRO-11 and C
modules structured into levels corresponding to local-net, IP,
TCP, and application levels, with user interfaces at each level
(Figure 1). The IP level conforms to the RFC-791
specification, including fragmentation, reassembly, extended
addressing, and options, but it does not interpret options. A
full set of ICMP features compatible with RFC-792 is available,
including destination-unreachable, timestamp, redirect, and
source-quench messages. Destination-unreachable and
source-quench information is conveyed to the user level via the
TCP and raw-datagram protocol modules. Internet gateway
(routing and non-routing) facilities compatible with IEN-109
(as amended) can be included on an optional basis. This
support can be configured to include hierarchically structured
gateways and subnets.

The TCP level conforms to the RFC-793 specification,
including PUSH, URGENT, and options. Its structure is based on
circular buffers for reassembly and retransmission, with
repacketizing on each retransmission. Retransmission timeouts
are dynamically determined using measured roundtrip delays, as
adjusted for backoff. Data flow into the network is controlled
by measured network bandwidth, as adjusted by source-quench
information. Features are included to avoid excessive segment
fragmentation and retransmission into zero windows. The user
interface level provides error and URGENT notification, as well
as a means to set outgoing IP/TCP options.

APPLICATION PROGRAMS

TELNET, FTP ETC.	SYSTEM MON & CTRL WATCH, PING	SYSTEM MAINT & DEVEL XNET

TCP PROTOCOL MODULE		RTP PROTOCOL MODULE	SYSTEMS PROGRAMS
INTERNET PROCESS			

NETWORK IN/OUT PROCESS			NETWORK INTERFACE
SYNCHRONOUS DRIVERS	ASYNCHRONOUS DRIVERS	1822 DRIVERS	

DUV11, ETC.   DLV11, ETC.   1822

Figure 1. System Architecture

A raw-datagram interface is available for XNET (IEN-158), UDP (RFC-768), and similar protocols. It includes internal congestion and fairness controls, multiple-connection management, and timestamping. Protocols above UDP supported in the present system include time server (IEN-142) and name server (IEN-116). A number of user-level protocol modules above TCP have been built and tested with other internet hosts, including user/server TELNET (RFC-764), user/server FTP (RFC-765), user/server SMTP (RFC-821), and various other file-transfer, debugging, and control/monitoring protocols.

4.2. System Support Software

The operating system in which the IP/TCP functions are embedded includes an emulator for RT-11, a DEC operating system intended for single-user real-time applications. There are two versions of the system: The Basic Operating System (BOS) provides a single RT-11 background process and multiple individually relocated foreground processes in PDP11/LSI-11 processors without memory-management hardware; the Virtual Operating System (VOS), provides multiple background processes in separate virtual spaces in PDP11/LSI-11 processors with memory-management hardware. Both systems can run standard RT-11 system utility and application programs and support standard RT-11 program-development software; however, neither system requires a DEC license, unless RT-11 editors, compilers, or utilities are required.

The BOS is designed for a single-user LSI-11/2 system with 30K words of memory using a background process and optional foreground processes. The background process can support standard RT-11 programs and network server programs, including TELNET, FTP and SMTP. A foreground process can support some network user programs including TELNET. Both processes run at the same time, allowing virtual-terminal access to remote systems at the same time as mail transfers, for example. The VOS is designed for multiple-user or multiple-application LSI-11/23 systems with up to 124K words of memory using a virtual background process for each user or application. Each virtual process can support the full set of RT-11 programs and network user/server programs.

Either system supports a number of direct-access storage devices, including floppy disks, fixed and moving-head disks, several types of communication interfaces, and special-purpose facsimile and speech devices. Disks that are currently supported include the DEC RX01/2, RL01/2, RK05, RJS03/04, PDT-150, and several off-brand models. Standard ASCII-compatible terminals are

supported along with downline loading for remote machines. In addition, the Peritek VCG-Q Display Interface (bit-map graphics), Dacom/Rapifax 450 Computerfax (digital facsimile transceiver), and Lincoln Laboratories LPCM (speech codec) are supported in a prototype multimedia message system.

Code sizes and speeds depend on the system configuration and features selected. A typical 30K-word LSI-11/2 single-user configuration with all features selected and including the operating system, device drivers, and all buffers and control blocks, leaves about 16K words for user-level application programs and protocol modules. A typical 124K-word LSI-11/23 configuration provides the same service for up to seven individually relocated users. Disk-to-disk file transfers across a DMA interprocessor link between LSI-11/23s operate in the 40-50 Kbps range with 576-octet packets. The 124K-word PDP11/34 INTELPOST adaptation supports two 56-Kbps lines and a number of lower-speed lines.

4.3. Network Support Software

The DCNET architecture supports virtual hosts and adaptive routing. A DCNET physical host is a PDP11-compatible processor that supports a number of *cooperating sequential processes*, each of which is given a unique port Identification. Every DCNET physical host contains one or more internet processes, each of which represents a virtual host identified by a unique host ID, which can be changed dynamically. Each DCNET physical host is identified by a unique host number to detect loops in the routing algorithm that establish the minimum-delay paths between the virtual hosts.

Each virtual host can support multiple internet protocols, connections, and a virtual clock. Each physical host contains a physical clock that can operate at an arbitrary rate and a 32-bit logical clock that operates at 1000 Hz and is assumed to be reset each day at 0000 hours UT. The routing algorithm provides for the synchronization of logical clocks throughout each DCNET clone and its subnets to within, typically, a few tens of milliseconds.

Each link to any of a set of DCNET subnets or foreign nets is associated with a pseudo-host that is assigned a fixed-host ID. Subnet or foreign gateways can connect and reconnect to these links in arbitrary ways. Within the DCNET itself, the links connecting the various hosts can be configured automatically in arbitrary ways, as long as the net remains fully connected. If full connectivity is lost due to a link or host fault, the virtual hosts in each of the surviving segments can continue to operate with each other and, once connectivity is restored, with all of them.

Datagram routing from a physical host to each of the virtual hosts in the net is determined by tables that contain estimates of round-trip delay and logical-clock offset for all virtual hosts in the net. These estimates are updated by periodic messages exchanged on the links connecting physical-host neighbors. The updating algorithm is similar to that formerly used in the ARPANET and contains provisions for loop avoidance and host up/down detection.

4.4. Protocol Configurations

At this time and with certain minor exceptions, the DCNET Fuzzballs support a common set of low-level and high-level protocols designed to provide virtual-terminal, file-transfer and electronic-mail utility services that are compatible with other hosts in the DARPA Internet system.

4.4.1. Low-Level Protocols

Low-level protocols are a permanent part of the BOS/VOS operating system and serve as an interface to user application programs. The application programs are dynamically loaded automatically in response to a request by a remote internet host or manually as specified by a local user. The following is a list of the low-level protocols supported by the present implementation:

o   Internet Protocol (IP): The IP supports all functions, but ignores IP options. Features include fragmentation, reassembly, and extended addressing.

o   Internet Control Message Protocol (ICMP): The ICMP is fully supported. Features include source-quench, unreachable, redirect, and timestamps.

o   Gateway-Gateway Protocol (GGP): The GGP supports all functions, including extended addressing, but not Host Monitoring Protocol (HMP).

o   Transmission Control Protocol (TCP): This supports all functions through a protocol module interface to user application programs. Features include dynamic retransmission timeouts, congestion controls, and URGENT.

o   Datagram Protocols (UDP, XNET, etc.), also called Real-Time Protocol (RTP): This supports all functions through a protocol-module interface to user application programs.

4.4.2. High-Level Protocols

High-level protocols are supported by a set of application
programs that are designed to run in ordinary user processes.
These programs obtain ordinary system and file services using
standard RT-11 programmed requests and a set of software modules
called the RT-11 Emulator. Both sequenced (TCP) and datagram
(RTP) access to the low-level protocol modules of the internet
system is available using the same mechanism. The following is a
list of the high-level protocols supported by the present
implementation:

o   Virtual Terminal Protocol (TN): User and server
    programs are compatible with a subset of RFC-764. The
    TELNET user program supports typescript functions and
    Tektronix 4000-series graphics using the Peritek VCG-Q
    bit-map color display. The server TELNET interfaces to
    the emulated RT-11 system to run ordinary system and
    user application programs. Features include option
    negotiation and interrupt.

o   File Transfer Protocol (FTP): User and server programs
    are compatible with RFC-765 and can be used to exchange
    ASCII and image data with other Internet server and user
    hosts conforming to this specification.

o   Network-Independent File Transfer Protocol (NIFTP):
    User program (only) can be used to exhange ASCII and
    binary data with other Internet server hosts conforming
    to the UK "Blue Book" specification.

o   Simple Mail Transfer Protocol (SMTP): User and server
    programs are compatible with RFC-822 and can be used to
    exchange and forward mail with other Internet server and
    user hosts conforming to this specification.

o   Operator Intercom (TALK): TALK IS supported by
    DCNET-specific user and server programs that link
    operators in foreign and local hosts via the operator
    terminal. It operates like TOPS-20 TALK facility, but
    does not require both operators to be on the same
    machine.

o   Cross-Net Debugging Protocol (XNET): User and server
    program can be used to download remote hosts, port
    expanders, and gateways and as a simulated "bootstrap"
    for the XNET user program on some ARPANET service hosts.

o   SIMP Monitoring Protocol (SMP): This server program can
    be used to decode, format, and display reports derived
    from the Atlantic SATNET SIMP monitoring messages.

o   Network Measurement and Debugging Utilities (PING):
    User/server program can be used to measure and display

one-way and roundtrip delays with cooperating internet
hosts, port expanders and gateways.

o   Name and Time Server Utility (NAME): This user/server
    program can be used to access remote name and time
    servers, display host-address mapping tables, and set
    the local clock from a remote host.

o   Other Utilities: These include server programs for
    echo, discard (sink), character-generator, and other
    similar testing utilities.

4.4.3. User Interface and Application Programs

Standard RT-11 langauge processors and system utilities
(including KED, PIP, FORTRAN and others) operate in a DCNET host
with the RT-11 Emulator in the same way they would in a native
RT-11 system. However, some of the features useful in the native
system are unavailable in the emulated system, including command
files and some peripheral devices.

Several user application programs have been constructed for
use in the BOS/VOS operating systems and also operate in the
native RT-11 system. The following are the most useful of these:

o   Mail Utility Program (MSG): The MSG operates much like
    the TOPS-20 program of the same name to display and copy
    messages in standard "mail file" format. Most features
    of the TOPS-20 program are supported in the present
    implementation with the exception of the "answer" and
    "forward" functions.

o   Message Editor Program (SNDMSG): This can operate
    either in a stand-alone mode or as part of MSG. It
    solicits header information from the user and formats
    the message in standard format. Features include
    "include address list" and "include file" functions.

4.5. DCNET Systems at LINKABIT and Elsewhere

The original DCNET prototype was built under a DARPA contract
at COMSAT Laboratories. Figure 2 shows its configuration, which
consisted of several hosts located at COMSAT headquarters in
Washington, D.C., and at COMSAT Laboratories in Clarksburg, MD.
A gateway to ARPANET was located in Washington and a gateway to
SATNET was located in Clarksburg. The net and its hosts were
used as a testbed for the development of the local-net protocols
and Internet support software as described in this overview and
in the final COMSAT report to DARPA.

4.5.1. DCNET at LINKABIT

The prototype network has since been dismantled and relocated to LINKABIT Eastern Operations in McLean, VA. Figure 3 shows the configuration proposed for early 1983. At present all LSI-11 hosts except DECNET-GATEWAY are operational and running the DCNET software described above. Pending completion of Ethernet software drivers, these hosts are interconnected via point-to-point links using ARPANET 1822 DMA interfaces. An additional LSI-11 host is connected via 1200-bps dial-up line to DCN-GATEWAY, together with other networks at Ford Motor Company and the University of Maryland. The VAX-11/730 is scheduled for delivery in early 1983 and will run a VAX/Unix software package including IP/TCP modules from BBN.

The DECNET-GATEWAY, under development by an internally funded LINKABIT research program, is to serve as a gateway between DCNET and the cluster of TOPS-20 and VAX-11 machines located at LINKABIT Headquarters in San Diego. It will be connected by an existing 9600-bps link and replace a statistical multiplexor now used on that link. LINKABIT hopes to install IP/TCP software (under development now by DEC) on the TOPS-20 machines and connect one of them directly to DCNET using a satellite link as part of another internally funded research program. The DECNET gateway will eventually support virtual-terminal, file-transfer, and electronic-mail functions between Internet hosts and DECNET hosts. As resources permit, these functions will be extended to use X.25 public-network links as well as DECNET links.
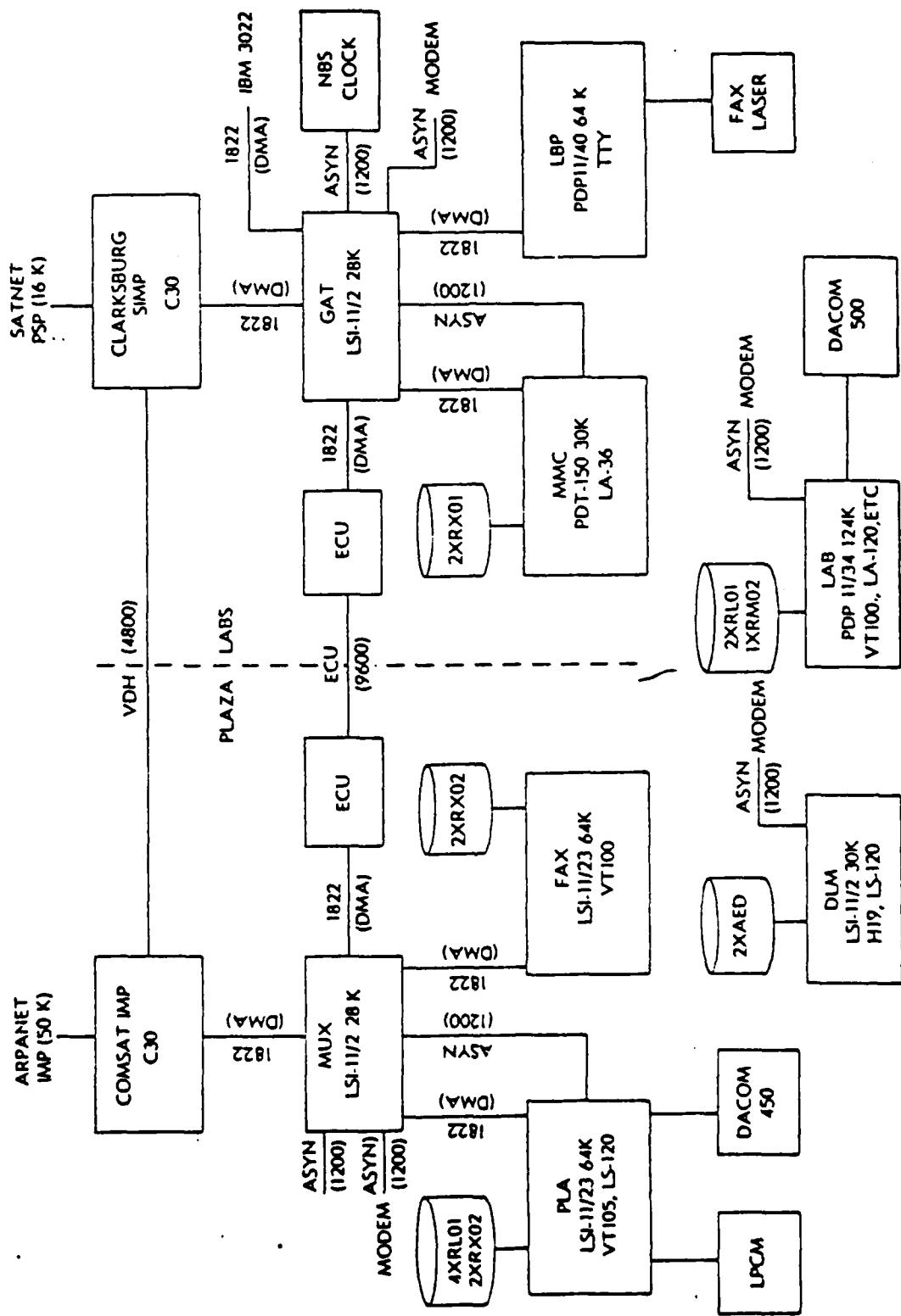
Figure 2. COMSAT DCNET Configuration
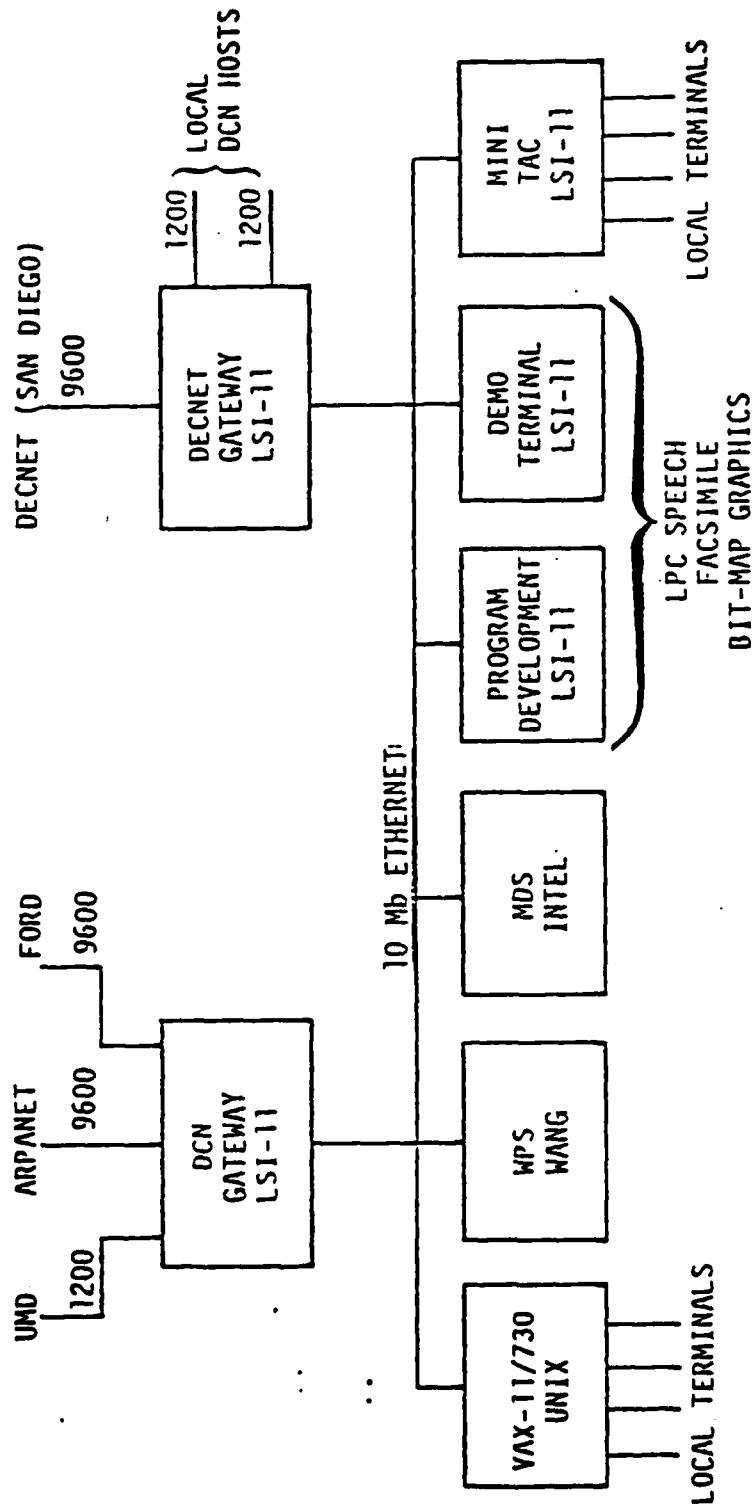
Figure 3. LINKABIT DCNET Configuration

The MINI-TAC represents what the ARPANET community calls a Terminal Access Controller (TAC) and what the public-network community calls a Packet Assembly/Disassembler (PAD). It is to function as a terminal concentrator and switching mechanism for all local terminals and hosts that are not accessible via the Ethernet. In principle, its function could be performed by the IDX switch if that device included the requisite protocol terminations, or by a suitably equipped LSI-11 host. An appropriate method has not yet been determined.

Figure 3 shows word-processing (Wang) and microprocessor-development (Intel) equipment connected to DCNET via the Ethernet. This is a speculative schematic; however, these and similar devices can be connected via serial asynchronous links and ad-hoc protocol modules.

4.5.2. DCNET Clones Worldwide

The DCNET architecture is in use for local nets in the United States and Europe and for international nets serving the United States, Europe, and South America. Figure 4 shows the known deployment of DCNET local nets connected to the DARPA Internet system. The Internet system employs three long-haul transport nets - ARPANET, SATNET and WIDEBAND (not shown) - providing connectivity throughout the United States and Europe. Not shown are the gateways that connect various local nets, shown as circles, to the transport nets, shown as ellipses. Hosts (either singly or in multiples) running DCNET software are shown as squares, with the prototype Internet gateway (developed as part of the LINKABIT-DARPA project) identified as "G".

Some of the local nets shown are structured as DCNET clones and use DCNET link-level protocols, as well as supporting software. In several cases the DCNET host software is used directly with other local-net architectures, including ARPANET, X.25, and contention-bus systems. DCNET clones include the following:
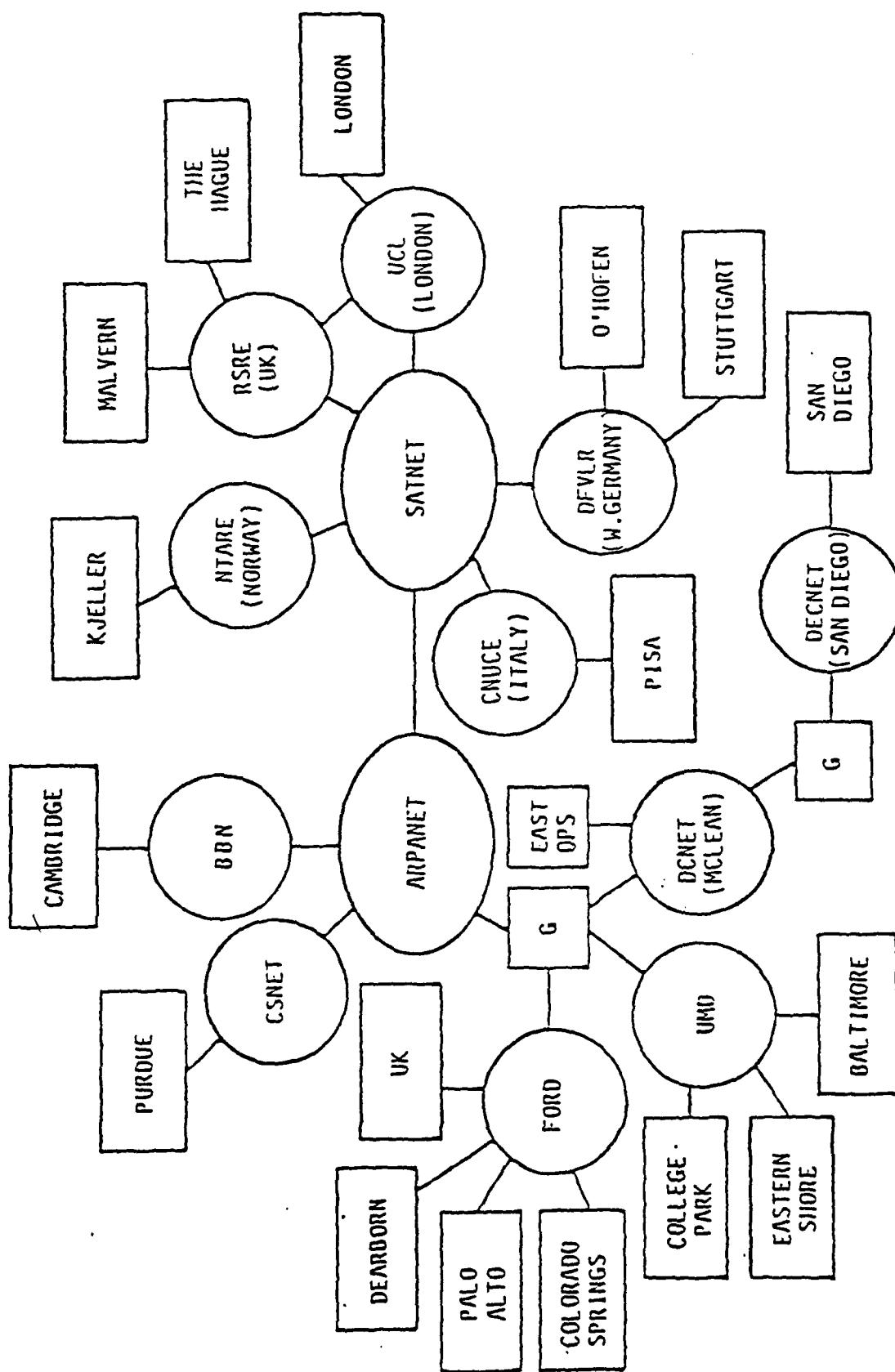
Figure 4. DCNET Clones Worldwide

o   FORD (Ford Motor Company, including Ford Aerospace).
    Ford has established a corporate-wide network based on
    the DCNET architecture, including DCNET clones located
    as shown and interconnected by 9600-bps long-haul links
    and contention-bus local-area links. Ford has
    constructed in-house or has contracted supporting
    software for several different host systems, including
    DEC, IBM, and Honeywell machines and operating systems.

o   INTELPOST (U.S. Postal Service and international
    counterparts) (not shown). The INTELPOST network, used
    for electronic-mail (facsimile) service between the
    United States, Europe, and South America, operates with
    DCNET protocols including IP/TCP and point-to-point
    links. The system uses a software package extracted
    from that developed for the DARPA Internet Program,
    together with high-level protocol modules developed for
    INTELPOST, and runs under the RSX-11 operating system.

o   UMD (University of Maryland). UMD has established a
    three-campus DCNET clone including LSI-11, VAX-11, and
    Univac hosts to be used for general academic and reseach
    functions.

o   NTARE (Norwegian Telecommunications Administration).
    This organization, in cooperation with the Norwegian
    MoD, has established a DCNET clone including two LSI-11
    hosts connected to the Tanum (Sweden) SATNET earth
    station. It is used primarily for research and
    demonstration.

o   DFVLR (West German counterpart of NASA). This
    organization has established a DCNET clone including
    three LSI-11 hosts to be connected to the Raisting
    SATNET earth station. Included on this network is a
    DARPA-owned LSI-11 host at Stuttgart. The net will be
    used for general research and demonstration, as well as
    for operational traffic between the Stuttgart host and
    domestic ARPANET hosts.

o   CNUCE (Italian counterpart of NASA/NSF). This
    organization has announced plans to establish a DCNET
    clone at Pisa to be connected to the Fucino SATNET earth
    station now on order from COMSAT.

4.6. Fuzzball Configurations

Fuzzball systems have been configured for a wide variety of
standard UNIBUS systems ranging from the PDP11/03 to the PDP11/70
and including memory to 256 kilobytes. Configurations for Q-BUS
components have been assembled using the LSI-11/2 and LSI-11/23
processors as well. The following is a set of guidelines that
may help in configuring new systems:

There are two standard configurations, one designed to support a single user using the LSI-11/2 processor and dual-floppy disk, and the other designed to support up to seven users using the LSI-11/23 processor and 30-megabyte Winchester disk. Either system can be distributed on two double-density diskettes. One diskette contains the files for the standard system, including certain files from the DEC RT-11 standard distribution. The other contains the files to generate special systems tailored for individual requirements. The entire set of sources necessary to generate the system can be distributed on four double-density diskettes.

Although it is, in principle, feasible to operate the system without any DEC RT-11 files, this is not recommended. These files provide system-utility functions, such as formatting diskettes, copying and editing files, and so forth. In addition, standard DEC RT-11 assemblers, compilers, and linkers are fully supported in the run-time system and are necessary for generating or reconfiguring the system under RT-11.

4.6.1. Standard Single-User Configuration

The standard single-user system includes the VT-103 video-display terminal, which serves both as the operator terminal and the power supply/backplane for the processor modules. The backplane is a 4 x 4 configuration and includes the following modules:

LSI-11/2 CPU with KEV11 EIS	MXV11-AC Multifunction
MSV11-DC Memory 32K bytes	RXV21 Disk Control
DRV11 Parallel Line Unit	SRI 1822 Interface
spare	spare

Bill of Material
1 VT103-AA LSI-11 Video Terminal
1 KD11-HA LSI-11/2 Processor Module
1 KEV11 EIS/FIS Chip for KD11
1 MSV11-DC Memory Module, 32K bytes
1 MXV11-AC Multifunction Module, 32K bytes RAM
1 MXV11-A2 Bootstrap PROM for MXV11
1 RXV21-BA Dual Floppy Disk Drive and Control Module
1 DRV11 Parallel Line Unit Module
1 SRI 1822 Interface Module for DRV11
1 SRI Cable Set for DRV11/1822 and 1822/IMP

Memory Map
```
000060   MXV11 line 1 interrupt vector (line 1)
000140   DRV11 (SRI 1822) interrupt vector
000264   RXV21 interrupt vector
000300   MXV11 line 0 interrupt vector (line 0)
176140   DRV11 (SRI 1822) control/status register
176500   MXV11 line 0 control/status register (line 0)
173000   MXV11 bootstrap PROM
177170   RXV21 control/status register
177560   MXV11 line 1 control/status register (line 1)
```

(Note: MSV11 strapped for 2K peripheral page)

The two MXV11 serial lines are connected to VT103 terminal ports, which provide connectors and speed selection. One of these (line 1) is used as the operator terminal, while the other (line 0) can be connected to a second terminal, a printer, or a modem. The standard configuration assumes this line is connected to a terminal or printer; however, this line can be configured as a DCN link to another system.

The standard system is configured for a single 16K-word user process, which can be used by the operator terminal or the network server, but not by both at the same time. Ordinarily, the operator manually enables the server function only when he is not actively using the system.

4.6.2. Standard Multiple-User Configuration

The standard multiple-user system includes the VT-103 video-display terminal, which serves both as the operator terminal and the power supply/backplane for the processor modules. The backplane is a 4 x 4 configuration and includes the following modules:

LSI-11/23 CPU with MMU	MSV11-LK Memory 256K bytes
DLV11-J 4 Serial Lines	DLV11-J 4 Serial Lines
ACC DMA Control	DSD 880 Disk Control
ACC 1822 Interface	spare
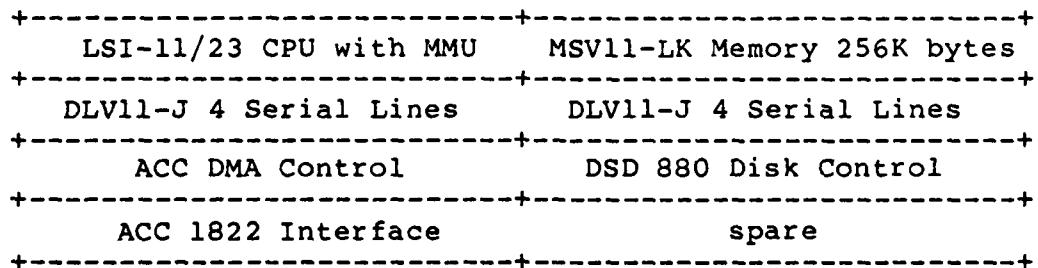
Bill of Materials
```
1  VT103-AA LSI-11 Video Terminal
1  KDF11-AA LSI-11/23 Processor Module with MMU
1  MSV11-LK Memory Module, 256K bytes
1  DLV11-J Serial Line Module, 4 lines
```

```
1   DSD 880-30 Winchester/Floppy Drive
1   ACC MDMA/1822 Control/Interface Module Set
1   ACC Cable Set for MDMA/1822 and 1822/IMP
```

Memory Map
```
000060   DLV11-2 line 3 interrupt vector (line 7)
000140   ACC 1822 interrupt vector
000160   DSD RL02 interrupt vector
000164   DSD RX02 interrupt vector
000300   DLV11-1 line 0 interrupt vector (line 0)
000310   DLV11-1 line 1 interrupt vector (line 1)
000320   DLV11-1 line 2 interrupt vector (line 2)
000330   DLV11-1 line 3 interrupt vector (line 3)
000340   DLV11-2 line 0 interrupt vector (line 4)
000350   DLV11-2 line 1 interrupt vector (line 5)
000360   DLV11-2 line 2 interrupt vector (line 6)
173000   DSD bootstrap PROM
174400   DSD RL02 control/status register
176200   ACC 1822 control/status register
176500   DLV11-1 line 0 contro/status register (line 0)
176510   DLV11-1 line 1 contro/status register (line 1)
176520   DLV11-1 line 2 contro/status register (line 2)
176530   DLV11-1 line 3 contro/status register (line 3)
176540   DLV11-2 line 0 contro/status register (line 4)
176550   DLV11-2 line 1 contro/status register (line 5)
176560   DLV11-2 line 2 contro/status register (line 6)
177170   DSD RX02 control/status register
177560   DLV11-2 line 3 contro/status register (line 7)
```

(Note: MSV11 strapped for 2K peripheral page)

Two of the DLV11 serial lines are connected to VT103 terminal ports, which provide connectors and speed selection. One of these (line 7) is used as the operator terminal, while the other (line 6) can be connected to a second terminal, a printer or a modem. The standard configuration assumes this line is connected to a terminal or printer; however, this line can be configured as a DCN link to another system. The remaining six DLV11 lines should be configured for speeds no higher than 4800 bps and can be connected to local or remote terminals as required.

There are seven virtual user processes configured in the standard system, six with 14K words of memory and the seventh with 16K words. The seventh user process is ordinarily used for the network server and is not available for terminal use, unless the server function is manually disabled. This configuration ordinarily supports the operator terminal together with five local or remote terminals and a single network server. The number of network servers can be specified at run time without shutting down the system.

4.6.3. Nonstandard Configurations and Options

There are combinations of processors, memories, and peripheral options too numerous to detail. The following is a summary of options useful in the above standard configurations. These fall into four categories: processor options, communications options, disk options, and special options.

4.6.3.1. Processor Options

The KWV11 Programmable Clock can be included in addition to the ordinary line-frequency clock. This provides increased stability for the timekeeping function; however, the line-frequency clock continues to provide interval-timer functions. The line-frequency clock also can be configured with or without a control/status register. Support is available for either the Spectracom or True-Time radio clock, which interface via serial line and require the KWV11.

MSV11 memories, including the memory-parity feature can be supported. If a memory-parity fault occurs in a user process, the program is notified by interrupt.

4.6.3.2. Communications Options

Either the SRI 1822 or ACC MDMA/1822 (or a mixture of both) can be included in either standard configuration. Up to four 1822 line cards can be supported on a single MDMA card. These devices and several others can be used to interconnect Fuzzballs on a DCNET local net. These devices include:

o   Serial asynchronous lines using various models of the DL11, DLV11 and DC11 line units at speeds up to 4800 bps. The protocol comforms to the DCNET specifications and uses character-stuffing and envelope conventions similar to IBM BSC.

o   Serial synchronous lines using the DU11, DUV11, DP11, DPV11 and DMV11 serial line units at speeds up to 4800 bps (56K bps for the DMV11). The protocol is similar to the above for all devices except the DMV11, which implements the DEC DDCMP.

o   Parallel lines using the DRV11 parallel line unit. An adapter box is required to connect two DRV11s back to back.

o   CSMA/CD Ethernet using the 3-COM and (non-Ethernet compatible) Ford LNA interfaces. (3-COM support is not yet complete.)

o   X.25 (level 2) using the RSRE serial line unit at speeds
      to 56K bps.

4.6.3.3. Disk Options

Several disks can be supported, either singly or in
combination. The disks that are currently supported include the
DEC RKV11/RK05 and DEC RLV21/RL01/RL02 cartridge disks, DEC
RH11/RJS03 fixed disk, DEC RXV11/RX01, DEC RXV21/RX02, PDT-150
and AED 6200 floppy disks, USDC CSS-1 and SMS Winchester disks,
as well as compatible disks of other manufacture.

4.6.3.4. Special Options

Support is available for digital facsimile storage and
retrieval at speeds up to 4800 bps. This requires the
Dacom/Rapifax 450 Computerfax and a DEC DUV11 synchronous line
unit. Support is also available for digital speech storage and
retrieval at 2400 bps. This requires Linear Predictive Codec
(LPC) equipment conforming to the LPC-10 specification and a DEC
DPV11 synchronous line unit. In addition, support is available
for bit-map display hardware using the Peritek VCG-Q interface.

APPENDIX   FUZZBALL USER INFORMATION

This Appendix consists of a set of descriptions of various
Fuzzball components that are of interest to the user and
applications programmer. The entries are sequenced
alphabetically in the same format used for the on-line HELP
facility.

BINCOM   Binary compare program

  SYNTAX
```
        BINCOM                     ;invoke BINCOM.SAV
        *[listfile][,SIPPfile]=oldfile,newfile[/options]
        *...                       ;* is CSI prompt
        *^Z                        ;exit to CLI
```

  SEMANTICS
        See the RT-11 System User's Guide

  OPTIONS
        /B   Compares bytes instead of words
        /D   Compares two entire volumes
        /E:n   Ends comparison at block n
        /H   Prints help information on terminal
        /O   Creates a difference listing file or SIPP command
                 file even if no differences between the input
                 files are found
        /Q   Suppresses terminal output of differences
        /S:n   Starts comparison at block n

  EXAMPLES
        None

BUP   Backup Utility Program

SYNTAX
        BUP   ;invoke BUP.SAV
        * filespec|[/options]= filespec|[/options]
        ...   ; is CSI prompt
        *^Z   ;exit to CLI

SEMANTICS
        See the RT-11 System User's Guide

OPTIONS
        /I   Backs up a large volume on multiple smaller
                    volumes
        /L   Lists directory of a backup volume
        /X   Restores a large file or, with /L, a large volume
                    from multiple backup volumes
        /Z   Initializes a volume for backup
no option   Backs up a large file on multiple smaller
                    volumes

EXAMPLES
        None

CRMAIL   Create Mail File

SYNTAX
```
     CRMAIL                    ;invoke CRMAIL.SAV
     | filespec|  blocks|      ;| is CRMAIL prompt
```

SEMANTICS

      This program creates a mail file with name filespec| and
      length blocks|. The file is initialized with the first
      byte set to SUB (^Z) and the remaining bytes set to zero.

OPTIONS

      None

EAAMPLES
```
     CRMAIL
     |unsent.msg 500
```

DIR   Directory Program

SYNTAX
        DIR   ;invoke DIR.SAV
        * filespec|[/options]   ;* is CSI prompt
        *...
        *^Z   ;exit to CLI

SEMANTICS
        See the RT-11 System User's Guide

OPTIONS
        The syntax of date is dd.:mmm:yy.   where dd-day,
        mmm-month, yy-year
        /A   Lists the directory alphabetically
        /B   Includes starting block numbers in directory
                  listing
        /C:n   Lists directory in n columns;   n can be 1 to 9
        /D:date   Includes only files with date
        /E   Lists entire directory, including unused spaces
        /F   lists short format directory in five columns
        /G   Lists directory entry of specified file and all
                  subsequent directory entries
        /J:date   Lists files created on or after date
        /K:date   Lists files created before date
        /L   Lists volume directory in order of entry
        /M   Lists unused areas
        /N   Lists directory summary
        /O   Gives sizes and block numbers in octal
        /P   Lists all files except those you specify
        /Q   Lists deleted files
        /R   Sorts directory in reverse order;   use with /S
        /S:xxx   Sorts directory listing;   xxx can be DAT, NAM,
                  POS, SIZ or TYP
        /T   Lists only protected files
        /U   Lists only unprotected files
        /V:ONL   Includes volume ID and owner name as part of
                  directory listing;   with ONL lists only ID and
                  name

EXAMPLES
        None

DUMP   File Dump Program

   SYNTAX
        DUMP   ;invoke DUMP.SAV
        * filespec|[/options]= filespec|[/options]
        ...   ; is CSI prompt
        *^Z   ;exit to CLI

   SEMANTICS
        See the RT-11 System User's Guide

   OPTIONS
        /B   Outputs octal bytes
        /E:n   Ends output at block n
        /G   Ignores input errors
        /N   Suppresses ASCII output
        /O:n   Outputs only block n
        /S:n   Starts output at block n
        /T   Defines a magtape as non-RT-11 file structured
        /W   Outputs octal words
        /X   Outputs Radix-50 characters

   EXAMPLES
        None

DUP   Device Utility Program

  SYNTAX
       DUP   ;invoke DUP.SAV
       * filespec|[/options]   ;* is CSI prompt
       *...
       *^Z   ;exit to CLI

  SEMANTICS
       See the RT-11 System User's Guide

  OPTIONS
       /B:RET   Writes FILE.BAD entries over bad blocks; use
                with /Z; with RET, retains FILE.BAD entries
                created on previous initialization
       /C   Creates a file; use with /G:n
       /D   Restores previously initialized volume
       /E:n   Specifies last block number; use with /I or /K
       /F   Prints names of bad blocks; use with /K
       /G:n   Specifies starting block number; use with /C,
                /I, or /K
       /H   Verifies after copying; use with /I
       /I   Copies image of one volume to another
       /K   Scans a device for bad blocks
       /N:n   Defines number of directoy segments; use with
                /Z; n can be 1 to 37 (octal)
       /O   Boots a volume or file (not supported in BOS/VOS)
       /Q   Boots a volume that is not RT-11 V4 or later;
                use with /O (not supported in BOS/VOS)
       /R:RET   Scans volume for bad blocks and creates a block
                replacement table; with RET retains previous
                table (not supported in BOS/VOS)
       /S   Consolidates free space on a volume
       /T:n   Extends a file by n blocks; n free blocks must
                follow the file
       /U:DEV   Writes bootstrap into blocks 0, 2 through 5 of a
                volume; DEV is the volume name (not supported in
                BOS/VOS)
       /V:ONL   Prints user ID and owner name; use with /Z to
                write new directory, ID and name on volume; with
                ONL, writes only the a ID and name
       /W   Waits for volume to be mounted before executing
                the command
       /X   Prevents automatic reboot after using /S on
                system device
       /Y   Suppresses query messages
       /Z:n   Initializes device directory; n is the number of
                extra words in each directory entry

EXAMPLES

None

FAX*   RAPICOM 450 Facsimile Operator's Guide

   SYNTAX
     SCANNING
          PIP   (* is CSI prompt)
          * filespec|=FAX:
          *...
          *^Z   (exit to CLI)
     PRINTING
          PIP   (* is CSI prompt)
          *FAX:= filespec|
          *...
          *^Z   (exit to CLI)

   SEMANTICS
          The RAPICOM 450 Computerfax Facsimile Transceiver scans
          and prints documents at 200-lpi resolution (1726 pel x
          2200 pel raster). It uses a two-dimensional adaptive
          run-length compression algorithm. See the ENCODE and
          DECODE programs to convert between this representation
          and CCITT T.4 or bit-map representation.

          Facsimile files can be transferred between Internet hosts
          using the FTP program (using IMAGE mode).

   OPTIONS
          None

   EXAMPLES
          ftp dcn2
          220 DCN2.ARPA FTP Service (16-Apr-83 Version) 21-May-83
          *login user password
          230 User user logged in at 21-May-83 21:04:03
          *get sy:startf.com x.y
          ?FTP-I-Transfer begins SY:STARTF.COM[1] to local
          DK:X.Y[1000]
          ?FTP-I-Transfer complete 1 blocks, 11 sec (372 bps)
          *send x.y
          ?FTP-I-Transfer begins DK:X.Y[73] from local DK:X.Y[1]
          ?FTP-I-Transfer complete 1 blocks, 6 sec (682 bps)
          *rename x.y y.z
          200 File DK:X.Y renamed DK:Y.Z
          *delete y.z
          200 File deleted DK:Y.Z
          *quit
          221 DCN2.ARPA Closing

FILES   File Specifications and Formats

SYNTAX
        logname|: filename|. extension|[size|]

SEMANTICS
        All programs operate with RT-11 file formats and either
        file-structured or sequential devices. RT-11 files are
        block-structured with a block size of 512 bytes. There
        are two types of files: ASCII and IMAGE. ASCII files
        consist of a stream of 7-bit ASCII characters stored in
        eight-bit bytes with the high-order bit set to zero.
        During transmission, the NUL and DEL characters are
        discarded, and the SUB character terminates the transfer.
        IMAGE files consist of a stream of eight-bit bytes with
        all bits significant. In both cases data are zero-filled
        to a block boundary before writing to the device.

        Files and devices are named according to RT-11
        conventions in the following syntax:

         logname|: filename|. extension|[size|]

        where logname| is the logical device name, consisting of
        up to three alphanumeric characters, filename| is the
        file name, consisting of up to six alphanumeric
        characters including $, extension| is the file
        extension, consisting of up to three characters including
        $, and size| is a decimal integer used to specify the
        maximum file size in blocks.

        The logical device name (together with the colon) is
        optional and defaults to DK:, which is the name of the
        standard RT-11 work volume. With certain exceptions
        transfers can be specified to or from either
        file-structured or sequential devices. In the latter
        case the name, extension and size are ignored. Logical
        device names are bound to physical devices by the CLI
        command ASG.

        The size| specification (together with the braces) is
        optional and applies only to new or replacement files to
        be stored on file-structured devices, not to files
        that are to be stored on sequential devices or to be
        read. The algorithm used to estabish the maximum size of
        a file is as follows: If the size| specification is
        missing or zero, half of the largest hole on the volume is
        allocated. If it is a negative number, then the entire
        extent of the largest hole is allocated. Otherwise, the
        number of blocks specified is allocated in the first hole

large enough to contain them.

OPTIONS
None

EXAMPLES
None

FRUN   Foreground Loader Program

SYNTAX
        FRUN filespec| ;invoke FRUN.SAV
        * filespec| ;* is CSI prompt
        ;exit to loaded program

SEMANTICS
        The foreground loader is used to load programs linked to
        run in an RT-11 foreground or in a system job with or without
        overlays. These programs usually have the .REL
        extension and contain relocation information along with
        the program text. This program would be used ordinarily
        only in systems without memory-management facilities and
        when one or more foreground processes have been
        configured in the system.

OPTIONS
        None

EXAMPLES
        None

FTP   File Transfer User Program

SYNTAX
>   FTP   ;invoke FTP.SAV
>   ...
>   QUIT   ;exit to CLI
>   ;or
>   FTP   hostname| ;invoke FTP.SAV and execute an
>   ;implied CONNECT   hostname|
>   ...   ;command
>   QUIT   ;exit to CLI

SEMANTICS

The File Transfer Protocol (FTP) functions are provided
using two programs: the FTP User and the FTP Server. The
FTP User runs in interactive mode on a DCNET host. The
FTP Server normally resides at some other host and is
activated upon receipt of a TCP connection on port 20.
The FTP User is controlled by user commands in the same
way as the Command Language Interpreter (CLI) and other
user programs. It in turn controls the FTP Server by
standardized server commands, with the actions taken
indicated by standardized replies. These programs
operate with the protocol described in RFC-765 and can be
used with other implementations conforming to this
protocol.

A file transfer operation proceeds by first opening a
control connection to a remote host using the CONNECT
command and then to its operating system using the LOGIN
command. In the case of the DCNET FTP Server, the LOGIN
sequence is supported, but the specified user name and
password are ignored. Following the connection phase,
the SEND and GET commands are used to transfer files to
and from the remote host as required. For each transfer,
a data connection separate from the control connection is
established by the remote server. The sender closes the
connection to signal end-of-file and terminate the
transfer. The QUIT command is used to close the control
connection and exit to the CLI.

The FTP User prompt is "*", which signifys that the program is
ready for the next command. The first prompt appears
when the herald is received from the FTP Server following
the CONNECT command. Subsequent prompts appear as each
command sequence is completed with the FTP Server.
Prompts will not appear following commands that have
only local significance or in which errors are detected.

FTP transfers can be in either ASCII (default) or IMAGE

mode. (see the FILES help information for relevant file
structures). ASCII mode is compatible with all
implementations of RFC-765 servers known at this time.
In the case of IMAGE mode with the TOPS-20, however, care
must be taken to assure that 8-bit bytes are stored four
per 36-bit TOPS-20 word. This is assured if the TYPE L 8
command is used, rather than the TYPE I command (see the
IMAGE command below).

OPTIONS
 ABORT
        If a file transfer is not in progress, do nothing. If it
        is, abort the transfer, send the ABOR command to the
        Server and wait for it to close the data connection.
 ASCII
        The ASCII and IMAGE commands set the mode for subsequent
        file transfers. The default is ASCII. In the case of
        the IMAGE command, a TYPE I command is sent to the server
        unless an optional argument is present, in which case a
        TYPE L bytesize| command is sent, where bytesize| is
        the argument. In all except unusual cases bytesize|
        should be 8. This feature is included to support the
        TOPS-20.
 ASG filespec|
        Assign the local work volume DK: for subsequent file
        transfers to filespec|. See the VOLUMES help
        information for further details.
 BRIEF
        The BRIEF and VERBOSE commands control output of detailed
        commentary dialog. BRIEF disables this and VERBOSE
        enables it. The default is BRIEF.
 CONNECT hostname|
        Open a control connection to host hostname|.
 CWD filespec|
        Assign the working directory on the remote host for
        subsequent file transfers to filespec|.
 DELETE remotename|
        Delete the file remotename| on the remote host. The
         remotename| must contain no more than 40 ASCII printing
        characters.
 DIRECTORY directory| localname|
        Open a data connection to the remote host previously
        specified by the CONNECT and LOGIN commands. Then
        transmit a list of the files in directory| on the remote
        host to localname| on the local host. The directory|
        must contain no more than 40 ASCII printing characters.
        When operating with the DCNET FTP Server, it must be a
        valid RT-11 file name and can include wildcards in the
        same manner as the RT-11 DIRECTORY command. The
         localname| must be a valid RT-11 file or sequential

device name.

DISCONNECT

If the control connection does not exist, do nothing.
Otherwise, send the QUIT command to the Server and wait
for it to close the connection.

GET   remotename| localname|

Open a data connection to the remote host previously
specified by the CONNECT, LOGIN and ASCII/IMAGE commands.
Then transfer the file remotename| on the remote host to
localname| on the local host. The remotename| must
contain no more than 40 ASCII printing characters, while
the localname| must be a valid RT-11 file name or
sequential device name. If localname| is missing,
assume the string remotename| in its place.

HELP

Display helpful information, including a list of
commands.

IMAGE

(see the ASCII command)

LOGIN   user| password|

Log into the remote host previously specified by the
CONNECT command as user| with password|. Both user|
and password| must contain no more than 40 ASCII
printing characters.

QUIT

Execute an implied DISCONNECT command and exit to the
CLI.

QUOTE ...

Send the remainder of the command line to the remote host
via the control connection.

RENAME   fromname| toname|

Rename the file fromname| on the remote host to
toname|. Both fromname| and toname| must contain no
more than 40 ASCII printing characters.

ROUTE   hostname1| hostname2| ...

Specify the Internet source route as the sequence
hostname1| hostname2| ... (up to nine).

SEND   localname| remotename|

Open a data connection to the remote host previously
specified by the CONNECT, LOGIN and ASCII/IMAGE commands.
Then transfer the file localname| on the local host to
remotename| on the remote host. The localname| must be
a valid RT-11 file or sequential device name, while the
remotename| must contain no more than 40 ASCII printing
characters. If remotename| is missing, assume the
string localname| in its place.

SHOW

Display a formatted summary of various quantities of
interest, including the current host, file and block
number.

VERBOSE
      (see the BRIEF command)

EXAMPLES
      None

FTPSRV   File Transfer Server Program

SYNTAX
     See RFC-765.

SEMANTICS
     The FTP Server is designed to be compatible with the
     protocol specified in RFC-765. It is invoked by the
     TELSRV program upon receipt of a TCP connection on port
     21. Commands consist of four characters, only the first
     three of which are significant. Replies consist of a
     three-digit code, the first of which indicates the status
     of the request according to the following codes:

     1xx   The action requested has begun. Additional
               server commands can be given to initiate
               unrelated actions.
     2xx   The action requested has been completed successfully.
     3xx   Additional commands are necessary to
               specify completely the action requested.
     4xx   The action requested was aborted due to an error
               condition believed to be temporary. The command
               may be tried again at a later time.
     5xx   The action requested was aborted due to an error
               condition believed to be permanent.

     FTP transfers can be in either ASCII (default) or IMAGE
     mode (see the FILES help information for relevant file
     structures). ASCII mode is compatible with all
     implementations of RFC-765 servers known at this time.
     In the case of IMAGE mode with the TOPS-20, however, care
     must be taken to ensure that 8-bit bytes are stored four
     per 36-bit TOPS-20 word. This is assured if the TYPE L 8
     command is used, rather than the TYPE I command (see the
     TOPS-20 documentation).

OPTIONS
 ABOR
     If a data transfer is not in progress, return a 200 reply
     code. If a data transfer is in progress, signal the data
     connection to close and return a 426 reply code, flushing
     data received meanwhile. When the connection closes
     return a 226 reply code.
 CWD filespec|
     Assign the work volume DK: on the local hosts for
     subsequent file transfers to filespec|. See the VOLUMES
     help information for further details.
 DELE filespec|
     Delete the file filespec| on the local host, where
      filespec| must be a valid RT-11 file name. Reply code

200 is returned.

HELP

Display helpful user information, including a list of FTP Server commands. All but the last line returned have reply code 111. The last line returned has reply code 211.

LIST   directory|

Open the data connection and transfer a list of files in directory| on the local host to the FTP User host. The directory| must be a valid RT-11 file name and can include wildcards in the same manner as the RT-11 DIRECTORY command. Reply code 150 indicates the transfer has begun and that a 2xx, 4xx or 5xx will be returned when the transfer terminates. The transfer terminates normally with a 226 reply code when the FTP Server closes the data connection and abnormally with a 4xx or 5xx reply code in all other cases. In the list of files returned each file name is on a separate line followed by size and date information.

MODE   code|

Set the transfer mode for subsequent data transfers to code|. Currently, code| must be S (STREAM), which is also the default. Reply code 200 is returned.

NLST   directory|

Open the data connection and transfer a list of files in directory| on the local host to the FTP User host. The directory| must be a valid RT-11 file name and can include wildcards in the same manner as the RT-11 DIRECTORY command. Reply code 150 indicates the transfer has begun and that a 2xx, 4xx or 5xx will be returned when the transfer terminates. The transfer terminates normally with a 226 reply code when the FTP Server closes the data connection and abnormally with a 4xx or 5xx reply code in all other cases. In the list of files returned each file name is on a separate line terminated by CR| LF| with no other information on the line.

NOOP

Do nothing. Reply code 200 is returned.

PASS   password|

Syntax check only (for compatibility). Reply code 230 is returned.

PASV

This command is used only in special situations involving a third-party transfer to indicate that the FTP Server should listen on a data connection and to return the host and port of this connection in the reply code. Note that the data connection is not opened at this time, but will be opened in passive mode as necessary by subsequent data-transfer commands. The reply code returned is:

227 Entering passive mode.   hl,h2,h3,h4,pl,p2

where hl,...,h4 is the host and pl,p2 is the port with
the same interpretation as the PORT command.

PORT hl,h2,h3,h4,pl,p2
This command is used only in special situations involving
a third-party transfer to specify the host and port for
the data connection. The host is specified by the four
decimal integers hl,...,h4 and the port by the two
decimal integers pl,p2. All six of these integers must
have values in the range 0-255. Note that the data
connection is not opened at this time, but will be opened
in active mode as necessary by subsequent data-transfer
commands. Reply code 200 is returned.

QUIT
Close the control connection and exit to the CLI. Reply
code 226 is returned.

REIN
Reinitialize the program status as it was upon initial
entry. Reply code 330 is returned along with the FTP
Server herald.

RETR filespec|
Open the data connection and transfer filespec| on the
local host to the FTP User host. The filespec| must be
a valid RT-11 file or sequential device name. Reply code
150 indicates the transfer has begun and that a 2xx, 4xx
or 5xx will be returned when the transfer terminates.
The transfer terminates normally with a 226 reply code
when the FTP Server closes the data connection and
abnormally with a 4xx or 5xx reply code in all other
cases.

RNFR filespec|
Begin a file-rename sequence specifying filespec| on the
local host as the existing file. The filespec| must be
a valid RT-11 file name. Reply code 333 is returned,
indicating that the RNTO command must follow.

RNTO filespec|
Conclude a file-rename sequence begun by the previous
RNFR command and specifying filespec| on the local host
as the new file. The filespec| must be a valid RT-11
file name on the same device specified by the previous
RNFR command. Reply code 200 is returned.

STAT
Display a formatted summary of various quantities of
interest, including the current host, file and block
number. All but the last line returned have reply code
114. The last line returned has reply code 214.

STOR filespec|
Open the data connection and transfer a file from the FTP
User host to filespec| on the local host. The

filespec| must be a valid RT-11 file or sequential
device name. Reply code 150 indicates the transfer has
begun and that a 2xx, 4xx or 5xx will be returned when
the transfer terminates. The transfer terminates
normally with a 226 reply code when the FTP User closes
the data connection, and abnormally with a 4xx or 5xx
reply code in all other cases.

STRU code|
Set the file structure for subsequent data transfers to
code|. Currently, code| must be F (FILE), which is
also the default. Reply code 200 is returned.

TYPE code|
Set the representation type for subsequent data transfers
to code|. Currently, code| can be A (ASCII), I (IMAGE)
or L (IMAGE). The default when this command has not been
given is A. Note that the specification permits an
additional argument; however, it is ignored by the
FTP Server. Reply code 200 is returned.

USER user|
Syntax check only (for compatibility). Reply code 331 is
returned, indicating that the PASS command must follow.

EXAMPLES
See RFC-765.

GENERAL   General Description

SYNTAX
Following is a summary of syntactic types
/options   RT-11 file specification options (see
                  OPTIONS in HELP information)
address| Internet address (four fields separated
                  by ",". Each field is a decimal integer in
                  the range 0 through 255)
argument| Argument list (strings separated by SP|
                  and terminated by CR|)
command| Command name (only the first three
                  characters are significant)
devname| RT-11 physical device name
filespec| RT-11 file specification (see HELP
                  information for FILES)
hid| Host ID (decimal integer in the range
                  0-n, where n depends on network)
hostname| Host name (registered in NIC data base
                  HOSTS.TXT, or four fields as in address|
                  format).
logname| RT-11 logical device name (assigned by
                  ASG command)
pid| Port ID (octal integer in the range 0-n,
                  where n depends on configuration)

SEMANTICS
The DCNET internet software system has been developed
with DARPA sponsorship over the last several years. The
system has been used extensively for testing, evaluation,
and experimentation with other implementations. It
currently runs in a sizable number of PDP11s and LSI-11s
with varying configurations and applications. The system
is designed to be used with the DCNET local network and
BOS/VOS operating system for a multi-media internet
workstation (called a "fuzzball"), which operates using
emulation techniques to support the DEC RT-11 operating
system and application programs. However, the system has
also been used on other networks, including ARPANET, and
with other operating systems, including RSX-11. An
RSX-11 based version incorporating only the IP/TCP
modules is presently used to support the INTELPOST
electronic-mail network.

The software system consists of a package of MACRO-11 and
C modules structured into levels corresponding to
local-net, IP, TCP, and application levels, with user
interfaces at each level. The local-net level supports
several communication devices, including synchronous and
asynchronous serial lines, 16-bit parallel links and 1822

interfaces. Hosts using these devices have been connected to ARPANET IMPs, Satellite IMPs, MACRO-11 Internet Gateways, SRI Port Expanders, and to the DCNET local network. When used on DCNET, the system provides automatic routing, time-synchronization, and error-reporting functions.

The IP level conforms to the RFC-791 specification, including fragmentation, reassembly, extended addressing, and options, but currently does not interpret options. A full set of ICMP features compatible with RFC-792 is available, including destination-unreachable, timestamp, redirect and source-quench messages. Destination-unreachable and source-quench information is conveyed to the user level via the TCP and raw-datagram protocol modules. Internet gateway (routing and non-routing) facilities compatible with RFC-823 can be included on an optional basis. This support can be configured to support hierarchically-structured gateways and subnets.

The TCP level conforms to the RFC-793 specification, including PUSH, URGENT, and options. Its structure is based on circular buffers for reassembly and retransmission, with repacketizing on each retransmission. Retransmission timeouts are dynamically determined using measured roundtrip delays, as adjusted for backoff. Data flow into the network is controlled by measured network bandwidth, as adjusted by source-quench information. Features are included to avoid excessive segment fragmentation and retransmission into zero windows. The user interface level provides error and URGENT notification, as well as a means to set outgoing IP/TCP options.

A raw-datagram interface is available for non-TCP protocols such as UDP (RFC-768). It includes internal congestion and fairness controls, multiple-connection management, and timestamping. Protocols above UDP supported in the present system include Time Server (IEN-142) and Name Server (IEN-116). Other raw-datagram services include XNET (IEN-158), GGP Gateway (RFC-823), along with developmental versions of an EGP Gateway (RFC-827) and a DECnet gateway.

A number of user-level protocol modules above TCP have been built and tested with other internet hosts, including user/server TELNET (RFC-854), user/server FTP (RFC-765), user/server SMTP (RFC-821), and various other file-transfer, debugging, and control/monitoring protocols.

Code sizes and speeds depend greatly on the system
configuration and features selected. A typical 30K-word
LSI-11/2 single-user configuration with all features
selected and including the operating system, device
drivers and all buffers and control blocks, leaves about
16K words for user-level application programs and
protocol modules. A typical 124K-word LSI-11/23
configuration provides the same service to a half-dozen
individually relocated users. Disk-to-disk FTP transfers
across a DMA interprocessor link between LSI-11/23s
operate in the range 30-50 Kbps with 576-octet packets.
The 124K-word PDP11/34 INTELPOST adaptation supports two
56-Kbps lines and a number of lower-speed lines.

OPTIONS
    None

EXAMPLES
    None

HELP   Lists Helpful Information

SYNTAX
        HELP   ;invoke HELP.SAV
        What topic do you want help with?   argument|
                                ;...? is prompt
                                ; argument| has the form: topic
                                ;[subtopic[:items...]...]] or *

SEMANTICS
        *   lists the items for which help is available.
         CR| lists the HELP text (of which this is a part).
        topic   lists information on the specific topic only.
        topic subtopic lists information on the specific subtopic
                only (for example, HELP HELP SEMANTICS lists the
                paragraph of which this text is a part).
        topic subtopic:item lists only the text associated with
                the specific item.
        topic/item lists the text associated with the specific
                item under the subtopic OPTIONS.
        Subtopics are "SYNTAX", "SEMANTICS", "OPTIONS", and
                "EXAMPLES". Items are specific command options.

OPTIONS
        None

EXAMPLES
        None

HOST   Host Process Commands

  SYNTAX
        SET   devname| command| argument| ;CLI command

  SEMANTICS
        See USER process commands and GENERAL help information

  OPTIONS
   CMD
        List the available commands. Additional information is
        available with the HELP program.
   HOST   hosnam| pid| delay| status|
        Set the Host Table entry for the DCNET virtual host
        hosnam| to the values specified in the arguments. The
        Network Process PID is specified by the first argument,
        the roundtrip delay by the second, and the status by the
        last. If hosnam| is missing, simply display a summary
        of the entire table. If hosnam| is given with no
        arguments, display the table entry for that host. See
        Appendix C for a list of DCNET virtual hosts. Note that
        the Host Table is constructed automatically during normal
        system operations and that this command is provided only
        for exceptional circumstances.
   IDENT hid| address|
        Set the virtual-host ID for the master-clock host to
        hid|. If address| is given, set the internet address
        for the physical host to the address|.
   MINIMUM lbound|
        Set the minimum source-quench threshold to lbound|.
        ICMP source-quench messages will be returned to the
        sender if the number of free buffers falls below this
        threshold.
   NET   netnam| arglst|
        Set the Network Table entry for the network netnam| to
        the values specified in arglst|. The only argument that
        can be specified at present is the virtual host HID for
        the gateway to that network. If netnam| is missing,
        display a summary of the entire table. If netnam| is
        given with no arguments, display the table entry for that
        network.
   OFF   code|
        The ON and OFF commands can be used to set and clear bits
        in the options word of the parameter area associated with
        the process. The ON command sets the option bits
        corresponding to the one bits of code| (which is an
        octal value) to one, while the OFF command sets these
        bits to to zero. Options for the host process are
        interpreted as the RT-11 configuration word, which is
        normally stored at relative location 300 in the monitor

```
        area (see RT-11 documentation).
  ON  code|
        (see the OFF command)
  SHOW
        Display a formatted summary of various quantitites of
        interest.
EXAMPLES
        .set hos cmd
        Commands are:
        CMD   SHO   ON   OFF   ID   MIN   NET   HOS

        .set hos hos
        Host    PortID   Delay   Offset   Status   Leader
        ---------------------------------------------------------
        DC6     002      0       0        119      000000000000

        .set hos net
        Net     Address        HostID   Hops   Leader
        -----------------------------------------------
        ARP     [10.0.0.0]     0        0      0,0
        DCN     [128.4.0.1]    6        0      0,0
        WAS     [128.4.1.2]    14       0      0,0
        SAN     [128.4.2.3]    11       0      0,0
        FRD     [128.5.0.4]    13       0      0,0
        UMD     [128.8.0.5]    12       0      0,0
        NET     [0.0.0.6]      0        255    1,20

        .set hos sho
        Process type:  000030 options:  111001
        Last clock update:  0 from:  1 reset:  00:00:00
        Processes:        18
        Vectors:          10
        Nibbles:          29
        Small packets:    0
        Large packets:    12
        Min packets:      2
        Internet address:  [128.4.0.6]
```

INFORMATION   General Information

  SYNTAX
        None

  SEMANTICS
   BOOTING
        Mount the DCNET work volume on DK:. Mount an ordinary
        RT-11 system volume on SY:. The system volume should
        contain at least the following files:

        PIP.SAV   Peripheral interchange program
        DUP.SAV   Device utility program
        DIR.SAV   Directory program
        KED.SAV   Keypad Editor

        and other system and application programs as desired.
        The work volume should contain at least the following
        files:

        CLI.SAV   Command language interpreter
        FRUN.SAV   Foreground process loader
        TN.SAV   TELNET virtual terminal user program
        TELSRV.SAV   TELNET virtual terminal server program
        FTP.SAV   File transfer user program
        FTPSRV.SAV   File transfer server program
        SMTP.SAV   Mail transfer user program
        SMPSRV.SAV   Mail transfer server program
        MSG.SAV   Mail reader program
        SNDMSG.SAV   Mail composition program

        The following files will be useful for many applications:

        NIFTP.SAV   Network-independent FTP user program
        PING.SAV   Internet measurements user/server program
        XNET.SAV   Internet loader/debugger user/server
                           program
        NAME.SAV   Internet name/time user/server program
        GATEn.SAV   Internet gateway (special configuration)
        NSP.SAV   DECnet gateway (special configuration)
        VCGSRV.SAV   Peritek VCG display server
        WATCH.SAV   SIMP monitor program
        CRMAIL.SAV   Mail-file utility program
        HELP.SAV   Help utility program
        HELP.TXT   Help file (this file)

        These files operate with all versions of the DCNET
        software, while MSG, SNDMSG, and CRMAIL operate with RT-11
        as well. In addition, the work or system volume should
        contain the DCNET resident system file configured for the

particular hardware system on which it is run. This file
and other RT-11 system components are used only to load
the DCNET resident system and are not referenced after
that.

Once the system and work volumes are mounted, load the
DCNET resident system using the RT-11 RUN command. Be
sure the real-time harware clock is running. The system
should come up with a message from TELSRV, indicating
that it is listening for a connection. One or more prompt
characters (".") may also appear. The system then begins
to send HELLO messages to neighboring hosts and constructs
its Host Table, which determines the routing to these and
other hosts on the network. This may take a minute or
two, during which time the SET HOST HOST command can
be used to identify the other hosts as they come up. The
SHOW command can be used to confirm the correct
configuration of the processes and devices in the system
and the SET HOST SHOW command to determine the resources
in use. Once the Host Table has stabilized, the system is
ready for use.

IN-CASE-OF-TROUBLE
In case of trouble, the following procedures may be
helpful. The system can be restarted at any time from
location zero, and then it will recompute the checksum of
the read-only code and data areas and reinitialize all
processes. In addition, an EOT (^D) function received
from an operator terminal will do the same thing, as will
the RESET command. Finally, the XNET Internet loader/
debugger can be used to send a special message to a
remote system, causing the same action. These mechanisms
are potentially hazardous, but in keeping with the uses,
the DCNET software is likely to be put. Protection
features may be incorporated into future versions.

CONFIGURATIONS
There are two versions of the DCNET resident system
presently in use. One of these, called the basic system,
supports up to 60K bytes of memory and PDP11 or LSI-11
processors without memory-management features. The
other, called the virtual system, supports the maximum
available memory and PDP11 or LSI-11/23 processors with
these features. The standard basic system is configured
for a 60K-byte LSI-11/2 with EIS and one 32K-byte
background process. The standard virtual system is
configured for a 256K-byte LSI-11/23 and up to seven
32K-byte virtual processes. The remainder of memory is
used for the resident system and various storage areas.

PROCESSES

Processes are identified by three-character physical
names, which appear in the SHOW and DEVICE command
arguments and their responses. In the case of ordinary
direct-access and sequential input/output devices, these
names correspond to standard RT-11 conventions. In many
cases, several instances of a particular process type may
appear, in which case the final character will be a digit
in the range zero through nine. Each process is
identified by a port identifier, or PID, which appears as
an argument in some commands. Following is a list of the
current process types (n denotes a digit):

```
HOS     Host process
DCn     Internet process
DHn     Network process (ARPANET 1822 link)
DHn     Network process (DMV11 link)
LHn     Network process (serial synch or asynch link)
LNn     Network process (Ford serial LNA link)
LPn     Network process (Ford parallel LNA link)
RSn     Network process (RSRE X.25 interface link)
UCn     Network process (UCL HDLC interface link)
TTn     Terminal process operator terminal/line printer
BGD     Background user process
FGD     Foreground user process
VMn     Virtual user process
DXn     Disk process (RX01 single-density floppy disk)
DYn     Disk process (RX02 double-density floppy disk)
RKn     Disk process (RK05 disk cartridge)
DSn     Disk process (RH11/RJS03 fixed-head disk)
DLn     Disk process (RL11/RL01 disk cartridge)
FDn     Disk process (AED 6200 double-density floppy
        disk)
CSn     Disk process (USDC Winchester disk/tape
        cartridge)
SMn     Disk process (SMS Winchester disk)
FAX     Dacom 450 Facsimile process
LPC     LPCM packet speech process
UDP     Internet name/time server process
GAT     Internet gateway process
NSP     DECnet gateway process
```

As in RT-11, logical names can be associated with most of
the above names using the ASG command.

VIRTUAL-HOSTS

Of fundamental importance to the DCNET architecture is
the concept of physical and virtual hosts. A physical
host is an ordinary processor with memory and a
complement of direct-access and sequential devices. A

virtual host is a portable process that resides in a
physical host. Virtual hosts are assigned names and
32-bit internet addresses as shown elsewhere. Each
virtual host residing in a physical host is supported by
an Internet process, the name of which is the name of the
virtual host. The internet address of the physical host
itself (necessary only to break routing loops) must be
the address of one of its virtual hosts.

Virtual hosts are identified within DCNET by the host
identifier (HID), which presently is taken as the fourth
octet of the four-octet internet address. The particular
virtual host supported by a given Internet process is
determined by its HID, which can be changed by an
operator command. When the HID is changed, the name is
changed automatically as well.

TERMINAL-OPERATIONS
Operation of the standard DCNET terminal is like RT-11,
but with some differences. Following is a list of the
special keyboard functions (codes correspond to the
standard ASCII interpretation):

Code	Echo	Function
CR	CR-LF	End of line
SUB	^Z	End of file
DEL	BS-SP-BS	Delete previous character
CAN	^U	Delete current line
ETX	^C	Interrupt
DLE	none	Begin escape sequence
EOT	^D	Panic reset (not in RT-11)
DC3	^S	Stop output
DC1	^Q	Resume output

The DCNET interrupt and end-of-file functions operate
rather differently than their RT-11 counterparts. A
single (not two) ETX function always causes an immediate
termination of the program. A SUB function causes the
emulated RT-11 Command String Interpreter (CSI) to return
to the DCNET Command Language Interpreter (CLI). The DLE
function introduces a special function sequence
consisting of the DLE itself followed by a single
character. It the following character is another DLE, a
single DLE is placed in the input buffer. If it is a
digit, the function performed is as follows:

0	Switch to channel 0
1	Switch to channel 1
2	Switch to channel 2

3   Switch to channel 3
4-9   Reserved

Operator terminals ordinarily are connected to user
processes on a one-to-one basis. Usually channel 0 for
each terminal is assigned the associated user process and
the remaining channels assigned for special purposes,
such as net servers. These assignments can be changed
with the ASG command. The options in effect for each
terminal can be specified with the SET command.

MEMORY-MANAGEMENT

In the basic system, the supervisor and all processes
share the identity virtual-physical address mapping. The
hardware vector area is followed by the working storage
allocated to the background process. This is followed by
the code and read-only storage areas for the resident
system, including the supervisor and all processes. Next
is the code and read-only storage areas for the RT-11
emulator, which is re-entrant, position-independent, and
shared by all user processes. Following this is the
parameter area containing tables and other information
that can be accessed by the SET and SHOW commands.
Finally are the buffers, state vectors, and stacks used by
the various components of the system. The working
storage for each foreground process is allocated as part
of its state vector.

The virtual system is structured in a similar way, except
that each user process has its own address mapping.
Physical memory is organized as in the basic system,
except that there is no working storage allocated to the
background or foreground processes. Instead, the working
storage for each user process is allocated in turn above
the areas described in the previous paragraph. The
supervisor and all processes except user processes
operate in kernel space with identity virtual-physical
mappings, except that segment 7 is mapped to the
input/output page and segments 5 and 6 are used as
windows to other spaces. Each user process operates in
user space with its own working storage mapped to virtual
space starting at location zero. Segments 6 and 7 of
each user process are mapped to the RT-11 emulator and
parameter areas, while segment 5 is used as a virtual
window to other spaces. those functions that require it.

Within each user process in both the basic and virtual
systems, memory is organized as in RT-11. In the
background process the chain area starts at location 500,
and the various "RMON areas" and emulator working storage

extends downward from the end of its allocated working
storage. In the foreground processes the chain area
starts at the first location of its allocated working
storage. Virtual user processes appear as background
processes. Features have been incorporated into the
DCNET system to support re-entrant and position-
independent load modules in both background and
foreground processes.

EXAMPLES
    None

INTERNET   Internet Process Commands

SYNTAX
        SET devname| command| argument| ;CLI command

SEMANTICS
        See USER process commands and GENERAL help information

OPTIONS
  CCB cid| code|
        Display a summary of connection data from the Connection
        Control Block for connection cid|. If cid| is missing,
        display all open connections. The optional code|
        arguments, TRA and TCP, select additional displays
        applicable only to TCP. TRA displays a received packet
        trace, while TCP displays packet and signal counts by
        type.
  CMD
        List the available commands. Additional information is
        available with the HELP program.
  IDENT hid|
        Change the virtual host ID for the process to hid|.
        After this is done, the name of the process will change
        accordingly and the new routing information will
        circulate in the network. Once changed, the ID should
        not be changed again for at least two minutes, in order
        to allow old routing information to dissipate and to
        prevent loops.
  OFF code|
        The ON and OFF commands can be used to set and clear bits
        in the options word of the parameter area associated with
        the process. The ON command sets the option bits
        corresponding to the one bits of code| (which is an
        octal value) to one, while the OFF command sets these
        bits to to zero. Options for the internet process are as
        follows:

        12   NBS clock (OFF: disable, ON: enable)
        13   Echo all datagrams (OFF: disable, ON: enable)

  ON code|
        (see the OFF command)
  SHOW
        Display a formatted summary of various quantities of
        interest.
  SIZE size|
        Set the maximum datagram size to size|, which must be a
        decimal integer in the range 20-576 (40-576 for TCP).

EXAMPLES
        .set dc6 ccb

```
Connection ID:  051240 protocol:  006 state:  000007
  CCB format:  004000 max size:  256 protocol flags:  041
  Local addr:  [128.4.0.6] 3072 Foreign addr:
[128.4.0.6] 23
  Rate:  60 delay:  1055 RTD rate:  327 RTD delay:  17
```

```
.set dc6 ccb 4 tcp
Catenet source quench:      0
Catenet unreachable:        0
Input packets received:     66
  bad format:               0
  bad checksum:             0
  connection reset:         0
  dropped:                  1
Accepted packets:           66
  null (ACK-only):          24
  text stored:              42
  duplicate:                0
  outside window:           0
SYNs processed:             1
FINs processed:             0
Error packets sent:         0
Control packets sent:       1
Text packets sent:          25
Retransmissions sent:       0
ACK-only packets sent:      43
Data avail sigs to user:    43
```

```
.set dc6 ccb 4 tra
Seq      ID       Start    Length    Window   Offset
-----------------------------------------------------
30005    30005    0        1         399      0
35905    35905    0        216       184      0
37888    37888    0        46        354      0
38955    38955    0        1         399      0
46288    46288    0        68        332      0
46772    46772    0        1         399      0
62172    62172    0        155       245      0
63372    63372    0        1         399      0
64969    64969    0        214       186      0
1200     1200     0        1         399      0
5183     5183     0        216       184      0
6966     6966     0        46        354      0
8050     8050     0        1         399      0
25616    25616    0        216       184      0
27533    27533    0        216       184      0
30150    30150    0        216       184      0
32250    32250    0        216       184      0
34350    34350    0        216       184      0
36450    36450    0        108       292      0
```

```
38050    38050    0        1        399      0
46283    46283    0        216      184      0
48133    48133    0        216      184      0

.set dc6 cmd
Commands are:
CMD    SHO    ON    OFF    ID    SIZ    CCB

.set dc6 show
Process type:  000026 options:  000000
Host ID:  6 max conn:  8 max size:   256
Input packets:        156
   timeouts:          0
   net signals:       0
   bad format:        0
   bad checksum:      0
   returned:          0
   dropped:           0
Control msgs:         0
Output packets:       158
```

KED   Keypad Editor Program

SYNTAX
        KED   ;invoke BKED.SAV
        * filespec|[/options]   ;* is CSI prompt
        *...
        *^Z   ;exit to CLI

SEMANTICS
        Termination of the Keypad Editor
        [GOLD] [COMMAND] Prompt is Command:
        EXIT [ENTER] to save current edit version
        QUIT [ENTER] to terminate without saving edit
        modifications

        To define page size, screen size, and margin wrap
        [GOLD] [COMMAND] Prompt is Command:

        SET PAGE [ENTER] to number of lines per page
                        (Default=Formfeed)
        SET SCREEN [ENTER] to 80/132 characters per line on
                        screen
        SET WRAP [ENTER] to line length for auto right margin

OPTIONS
        /A:n   Allocates n blocks for output file
        /C   Creates a new file
        /I   Inspects input file; does not allow changes
                  No switch to edit file. A back-up file will be
                  created with the extension .BAK

EXAMPLES
        None

LIBR   Librarian Program

   SYNTAX
         LIBR   ;invoke LIBR.SAV
         *[libraryfile][,listfile]=inputfiles[/options]
         ...   ; is CSI prompt
         *^Z   ;exit to CLI

   SEMANTICS
         See the RT-11 System User's Guide

   OPTIONS
         /A   Includes in library directory all global symbols
                   includeing absolute global symbols
         /C   Allows multiple input lines
         /D   Deletes a module from a library file
         /E   Extracts a module from a library and stores it as
                   a .OBJ file
         /G   Deletes a global symbol from a library directory
         /M:n   Creates a macro library from an ASCII input file
                   and allocates n blocks for the macro name
                   directory
         /N   Includes module names in library directory
         /P   Includes psect names in library directory
         /R   Replaces modules in a library file
         /U   Updates (inserts and replaces) modules in a
                   library file
         /W   Produces a wide (132 column) library directory
                   listing
         /X   Creates a library with multiple global
                   definitions
         //   Allows multiple input lines until next occurrence
                   of //
      no option   Assumes module insertion

   EXAMPLES
         None

```
LINK              Linker program
```

SYNTAX
```
     LINK                      ;invoke LINK.SAV
     *[binfile][,mapfile][,stbfile]=objfiles[/options]
     *...                      ;* is CSI prompt
     *^Z                       ;exit to CLI
```

SEMANTICS
```
     See the RT-11 System User's Guide
```

OPTIONS
```
     /A        Lists global symbols in alphabetical order
     /B:n      Sets bottom address of program to n;  invalid
               with /H and /R
     /C        Continues input on new line;  do not use with //
     /D        Allows duplicate library subroutines
     /E:n      Extends root program segment to specified value
     /F        Uses default FORTRAN library FORLIB.OBJ when
               linking
     /G        Increases size of linker's library directory
               buffer
     /H:n      Specifies highest address to be used by
               relocatable code;  invalid with /B, /Q, /R, /Y
     /K:n      Inserts value of n into word 56 of block 0 as
               virtual SETTOP high limit;  n can be 1 to 32
               (decimal);  valid only with /V
     /L        Produces output file in .LDA format;  invalid
               with /R, /V
     /M:n      Defines stack address
     /N        Produces global cross-reference listing as part
               of load map
     /O:n      Produces overlay structure;  invalid with /L
     /P:n      Changes amount of space that linker uses for library
               routines;  default is 170
     /Q        Specifies start address of up to eight root
               program sections;  invalid with /R, /H
     /R:n      Produces output in .REL format;  n is the stack
               size;  invalid with /B, /H, /K, /L, /Q
     /S        Allows maximum memory space for linker symbol
               table
     /T:n      Defines transfer address
     /U:n      Rounds up a program section;  n must be a power
               of 2
     /V        Enables special XM monitor SETTOP and LIMIT
               features;  invalid with /L
     /W        Produces a wide load map listing (132 columns)
     /X        Does not output bitmap if codes is below 400
     /Y:n      Starts a program section on address boundary n;
               invalid with
```

LOGIN   Assign User Private Volume

SYNTAX
        login   usernam

LOGIN   Assign User Private Volume

SYNTAX
```
      login   username|           ;entered by user
      Password:    enter user password| ;host prompts for pw
                                  ;or
      login                       ;entered by user
      Login:    enter user name|  ;host prompts for name
      Password:    enter user password| ;host prompts for pw
```

SEMANTICS

Users of DCN encounter the login procedure in two
situations: When assigning a private virtual volume as
the default device, or when using the virtual-terminal
capability (TN) to use one host from another. In the
first case, the usage is voluntary depending on whether a
user wishes to use a private volume. If a private volume
does not exist; if the user name and/or private volume
are not registered in the login file; or if the system
default is adequate (the default volume (DK:) can be
determined with the DEVICE command), then the user might
choose not to log in. (See the help information for
SYSMGR if you wish to register a user name in the login
file.) In the second case, logging in is not voluntary;
rather it represents control of access to the DCN hosts,
as well as a method for assigning private volumes.

OPTIONS

None, but see also the help information for LOGOUT.

EXAMPLES
```
      .login smith
      .Password:
```
(Note: the password is not echoed, and a message is given
only in the event of an error.)

LOGOUT   De-assign User Private Volume

SYNTAX
       logout   ;entered by user

SEMANTICS
       After a user has assigned a private volume using the
       LOGIN command, the default volume may be restored by
       typing LOGOUT.

OPTIONS
       None, but see also the help information for LOGIN.

EXAMPLES
       None

MESSAGES   Message and Mail-file Formats

SYNTAX
     See RFC-822.

SEMANTICS

     The format of the ARPANET message data structures is
     described in RFC-822. In this model messages are sent by
     a user to a specified recipient in the format
     username|@ hostname|, where hostname| is the name of a
     host and username| is the name of an user known to that
     host. The implied address, usually called a mailbox, is
     typically associated with a mail file belonging to the
     recipient. The mail file format used in the DCNET system
     is line-structured, with each line terminated by the
     ASCII sequence CR| LF|, and contains only ASCII printing
     characters and format effectors. Messages consist of a
     file header, which contains a character count, followed
     by the message itself. Messages are stored one after the
     other, with the last followed by an ASCII SUB| character
     for compatibility with other RT-11 components. Figure 1
     shows the format of a typical message.

     17-Sep-83 17:53:49,314;000000000000
     MRCP to: Zorica@dcn5|
     MRCP to: @dcn1:Gross@dcn5|
     DLVD to: Mills@dcn6|
     Return-path: Mills@dcn6|
     Date: 17-Sep-83 17:53:25-UT
     From: Mills@dcn6
     Subject: Test message
     To: Zorica@dcn5
     cc: @dcn1:Gross@dcn5|, Mills@dcn6

     Folks,

     This message demonstrates RFC-821 and RFC-822 formats.

     Dave

     The first line is the file header. It includes the date,
     time, and count of characters in the message text and is
     followed by an array of twelve flag characters used by
     other programs of the message system. The message itself
     begins immediately following the CR| LF| that
     terminates this line and includes first the transport
     (RFC-822) header, followed by the message (RFC-821) header,
     and finally the text of the message itself. In the

above example, the lines beginning with MRCP and DLVD
belong to the transport header, and the lines following
that up until the blank line belong to the message
header.

Mail files can be created in three ways: (1) By copying
a mail file from a TOPS-20 or DCNET host as-is to a DCNET
host, (2) by creating and appending a new message locally
and (3) by receiving and appending a message from another
host.   Method (1) has been useful during testing and
in cases involving large amounts of mail that can be
bulk-transferred more efficient file-transfer protocols
like FTP and NIFTP.   However, TOPS-20 files do not
include the transport header, so that messages sent from
these files require manual intervention.   Method (2) is
implemented by an interactive mail editor, which operates
much like the TOPS-20 SNDMSG program to construct and
edit messages and append them, along with their transport
and message headers, onto a specified mail file.   Method
(3) is implemented by the SMTP Server, which listens for
messages from the network and appends them onto the
UNSENT.MSG mail file.

OPTIONS
     None

EXAMPLES
     See above.

MSG   Mail Reader Program

SYNTAX

        MSG   ;invoke MSG.SAV
        ...   ;-| is prompt
        Q(uit) or E(xit) or ^Z   ;return to CLI

SEMANTICS

        This module is part of the group of modules
        (SNDMSG,CRMAIL,SMTP) that form the mail system. The
        purpose of MSG is to examine and operate on messages that
        are stored in the mail file.

        The program first attempts to open a given file that must
        be in the mail format (created by CRMAIL). If it
        succeeds, it creates a table that holds the information
        about each message. The information is : Offset to that
        message, and number of bytes in that message. After
        displaying the information about the file blocks:

        A maximum of 500 file blocks are available of which 70
        are in use.

        it displays a prompt ' - '. The process is now ready to
        accept the user command. Every command is followed by a
        message sequence, which is composed of one of the
        following:

        1) a number
        2) two numbers separated by any non-numeric character
        3) one of the following special sequence characters:
                A - All messages
                D - Deleted messages
                E - Examined messages
                I - All messages in inverse order
                N - Not examined messages
                U - Undeleted messages

        The valid command characters and their operations are:

        A -   Answers to the specified message
        The process first displays the header of the message-
        Answer message:
        1   09-Mar Mills at dcn3   COMSAT final
        report(298char)
        Then it chains to the SNDMSG process to form the answer
        message. Below are the prompt fields issued by the
        SNDMSG process:

        From: enter user address

Reply-to (t or c): - Request for the address that will
appear on the 'To:' line in the answer message. In every
case, the address from the 'From:' field in the original
message (the senders address) is entered on the 'To:'
line. The following are the options for the 'To:' line:
[RET]   - only the senders address
t   - senders address and all the address in the
           'To:' field of the original message
c   - senders address, all the addresses in the
           original 'To:' field and all the addresses in the
           original 'Cc:' field.
Cc: - Additional address
Message:- text of the message

^Z terminates the message; the program then chains
back to the MSG. Note that the user is not prompted for
the subject. The answer messsage uses the same subject
title that was in the original message prefixed by 'Re:'.
The only time the user is prompted for the subject is
when the original message does not have that field. (For
more information check SNDMSG.)

B -   Backup, print previous message. It prints the
         message in the same format as in the t command

C -   Print the information about the current message
         number and file name:

Current message number is 36 of 37 messages in the file
UNSENT.MSG. A maximum of 500 file blocks are available
of which 70 are in use.

D -   Mark a given message sequence for deletion. The
         message remains part of the file until the file is
         updated.

E -   Exit the program and update the open file.
         Examines all of the messages and deletes the ones that
         are marked for deletion.

F -   Forward one or more messages. The process first
         displays headers of all the messages that have to be
         forwarded:

Forward message(s):
1   09-Mar Mills at dcn3   COMSAT final report

After this, the process chains to SNDMSG, which will build
the message that has to be forwarded. (Check SNDMSG for
explanation of prompt fields.) The first part of the
message text is the text that the user types in. The
second part is the inserted forwarded message(s).
Forwarded text is preceeded by a line that says-

--- beginning of forwarded message ---

and ended by the line:

--- end of forwarded message ---

^Z will complete the message and chain back to MSG.

G -   Set the current message number to the one given.

H -   Print the header information of the given message
        sequence. Prints the message number, date, from
        filed and the first 30 characters of the subject field:

1   09-Mar Mills at dcn3   COMSAT final report

I -   Include the byte count in the headers.
        It is an ON/OFF switch giving either a prompt,
        length included, or length NOT included.
        In the present implementation the length is always
        included.

L -   Create a listing file of the given message
        sequence. After entering the l command with the
        message sequence the process prompts for the file name:

List file name:
and then copies all the required message into the given
file.

N -   Next; print next message.
        Same action as the t command.

P -   Put a message from the open file into another
        file. The file created of the specified subset
        of messages is again in the mail file format.

Q -   Quit the MSG program.
        Quits the MSG program, closing the mail file without

updating it.   (See E command.)

S -   Chain to SNDMSG.
          Chains to SNDMSG process inorder to create a new message
          (for more information see SNDMSG)

R -   Use the given file as the active file.
          It opens a message file. If another message file was
          already active, it closes that file and issues a prompt:

Mail file name:
The new file becomes an active mail file

T -   Displays a given message sequence to the console.

U -   Unmark a given message sequence for deletion.

-         Mark a given message sequence as examined.

- -   Mark a given message sequence as not examined.

! -   Search the file "UNSENT.TXT".

? -   Type a list of valid command characters to the
          console. Prints the text from the msg.hlp file.

OPTIONS
     None

EXAMPLES
     None

NEC*   NEC Spinwriter Operator's Guide

SYNTAX
```
     PIP                        ;invoke PIP.SAV
     *LP:=NEC.                  ;* is CSI prompt
     *...
     *^Z                        ;exit to CLI
```

SEMANTICS
 SETUP
```
     Power-up printer
     Set characters/inch.  on switch SP to 10
     PIP[RET]
     *LP:=NEC.[RET]
     *^Z
     Select lines/inch and characters/inch on printer
     For text files without page size defined
     SET TTp PAGE nn[RET] p - LP:  terminal number, nn -
     lines/page
```
 PRINTING
```
     To set left margin use LM0, LM2, LM4, LM6 or LM8
     To set page size use PG72 for 6 lines/inch and PG96 for 8
     lines/inch
     RUNOFF[RET]
     *LP:=PG72,LM0,FILENM.RNO for 6 lines/inch and left margin
     at 0
     *^Z
     For text files with formfeed 'FF' for pages
     PIP[RET]
     *LP:=FILENM.EXT
     *^Z
```

OPTIONS
     None

EXAMPLES
     None

NETWORK   Network Process Commands

SYNTAX
        SET devname| command| argument| ;CLI command

SEMANTICS
        See USER process commands and GENERAL help information

OPTIONS
 CMD
        List the available commands. Additional information is
        available with the HELP program.
 IDENT address|
        Set the address assigned to the link to address|. that
        these values are constructed automatically during normal
        system operations and that this command is provided only
        for exceptional circumstances.
 OFF code|
        The ON and OFF commands can be used to set and clear bits
        in the options word of the parameter area associated with
        the process. The ON command sets the option bits
        corresponding to the one bits of code| (which is an
        octal value) to one, while the OFF command sets these
        bits to zero. Options for the network process are as
        follows:

        10   Link enable (OFF: enable, ON: disable)
        11   Loop detect (OFF: enable, ON: disable)
        13   External loopback (OFF: disable, ON: enable)
        14   HELLO messages (OFF: disable, ON: enable)

 ON code|
        (see the OFF command)
 SHOW
        Display a formatted summary of various quantities of
        interest.
 SIZE size|
        Set the maximum fragment size for outbound datagrams to
         size|. If size| is missing or is greater than the maximum
        permitted by network, assume the maximum (usually 256
        octets). Note that the size ordinarily is set at the
        maximum and that this command is provided only for
        exceptional circumstances.

EXAMPLES
        .set 1h0 cmd
        Commands are:
        CMD SHO ON OFF ID SIZ STA

        .set 1h0 sho

```
/H
/Z:n      Sets unused location to n
//        Allows multiple input lines;  use on first and
          last lines of input
```

EXAMPLES
    None

```
Process type:  000027 options:  040000
Subnet:  DCN status:  0 min buf:  0 retry:  3750
Foreign address:  [0.0.0.0] max size:  256
Input packets:      0
  bad format:       0
  bad checksum:     0
  returned:         0
  dropped:          0
  HELLO msgs:       0
ICMP msgs:          0
Output packets:    29
Input errors:       0
Output errors:      0
No buffer:          0
```

NIFTP   Network-independent File Transfer Program

    SYNTAX
        NIFTP   ;invoke NIFTP.SAV
        ...
        QUIT   ;exit to CLI

    SEMANTICS
        The Network-Independent File Transfer Protocol (NIFTP)
        has been the object of an ongoing standardization effort
        in the United Kingdom. The current specifications are
        given in a document called the Blue Book published by
        the National Physical Laboratory in 1980. A subset of
        this protocol is supported for file transfers between
        DCNET hosts and certain TOPS-20 ARPANET hosts using two
        programs, the NIFTP User, a PDP11 program written in the
        C language by Steve Treadwell, and the NIFTP Server, a
        TOPS-20 program written in the BCPL language by Chris
        Bennett, both of University College London.

        The NIFTP User runs in interactive mode on a DCNET host.
        The NIFTP Server resides at a TOPS-20 ARPANET host
        (presently USC-ISIE) and is activated upon receipt of a
        TCP connection on port 47. The NIFTP User is controlled
        by user commands in the same way as the Command Language
        Interpreter (CLI) and other user programs. It in turn
        controls the NIFTP Server by the protocol specified in
        the Blue Book. These programs can also be used with
        other implementations conforming to the Blue Book
        protocol.

        A file transfer operation proceeds by first specifying
        the remote host using the CONNECT command, and then the
        user name and password using the LOGIN command.
        Following this, the SEND and GET commands are used to
        transfer files to and from the remote host as required.
        For each transfer a connection is established to the
        NIFTP Server, which verifies the user name and password
        and establishes the working directory, following which
        the file transfer takes place. At the conclusion of the
        file transfer the connection is closed. The QUIT command
        is used to exit to the CLI.

        The NIFTP User normally controls all connection and
        file-transfer operations. The syntax of NIFTP User
        commands is similar to that of CLI commands and consists
        of a keyword followed by a list of arguments separated by
        spaces. Only the first three characters of a keyword are
        significant. In the case of some commands expecting two
        file names as arguments, if the second argument is

missing a copy of the first argument replaces it.
Ordinarily this convention leads to the intended action;
however, there are cases where misdirected use of this
feature can result in unintended actions.

The NIFTP User prompt is "*", signifying the program is
ready for the next command. A prompt appears as each
command sequence is completed with the NIFTP Server.
Prompts will not appear following commands that have
only local significance or in which errors are detected.

OPTIONS
 ABORT
        If a data transfer is not in progress, do nothing. Else,
        send the NIFTP Server ABORT signal and wait for the
        connection to be closed from the remote end.
 ASCII
        The ASCII and IMAGE commands set the mode for subsequent
        file transfers. The default is ASCII.
 BRIEF
        The BRIEF and VERBOSE commands control output of detailed
        commentary dialogue. BRIEF disables this, and VERBOSE
        enables it. The default is BRIEF.
 CONNECT   hostname| port|
        Set the hostname| and port| for a subsequent file
        transfer. The hostname| specification is in the form
        n1,n2,n3,n4, where n1,...,n4 are decimal integers with
        values in the range 0-255, and the default is the ARPANET
        host USC-ISID (10,1,0,27). The port| specification is a
        decimal number in the range 0-32767 with a default of 47
        (NIFTP).
 GET   remotename| localname|
        Open a connection to the remote host previously specified
        by the CONNECT, LOGIN, and ASCII/IMAGE commands. Then
        transfer the file remotename| on the remote host to
         localname| on the local host. The remotename| must
        contain no more than 40 ASCII printing characters, while
        the localname| must be a valid RT-11 file name or
        sequential device name. If localname| is missing,
        assume the string remotename| in its place.
 HELP
        Display helpful information, including a list of
        commands.
 IMAGE
        (see the ASCII command)
 LOGIN user| password|
        Specify the user| and password| for subsequent data
        transfers. Both user| and password| must contain no
        more than 40 ASCII printing characters.
  ..:T

Execute an implied ABORT command and exit to the CLI.
SEND localname| remotename|
Open a connection to the remote host previously specified
by the CONNECT, LOGIN, and ASCII/IMAGE commands. Then
transfer the file localname| on the local host to
remotename| on the remote host. The localname| must be
a valid RT-11 file or sequential device name, while the
remotename| must contain no more than 40 ASCII printing
characters. If remotename| is missing, assume the
string localname| in its place.
SHOW
Display a formatted summary of various quantities of
interest, including the current host, file and block
number.
VERBOSE
(see the BRIEF command)

EXAMPLES
None

PING   Internet measurements program

    SYNTAX
        PING   ;invoke PING.SAV
        ...
        QUIT   ;exit to CLI

    SEMANTICS
        The Packet InterNet Groper (PING) is an internet
        measurement and debugging tool. It can function both as
        a user and server and normally is run in interactive mode
        by a user at a DCNET host. PING is designed to send
        datagrams to cooperating internet hosts and gateways,
        which echo these datagrams to the sender. The roundtrip
        delays are then recorded and used to develop summary
        statistics and histograms. PING is not designed to test
        throughput capabilities, since only a single datagram is
        allowed on the roudtrip path. Ordinarily, PING datagrams
        use the Internet Control Message Protocol (ICMP) Echo and
        Echo Reply messages, since these are echoed by many
        Internet hosts and gateways, including all those using
        the DCNET internet software. However, the ICMP datagram
        formats can be used with other protocol numbers in the IP
        header.

        A PING operation proceeds by first opening an RTP
        datagram connection with the remote host using the
        CONNECT command. The PING command is used to begin a
        volley of ICMP Echo - ICMP Echo Reply exchanges between
        the local and remote hosts. Roundtrip delay samples are
        recorded in a buffer capable of holding about 500
        samples. Finally, the SHOW command is used to select the
        format and display the results. The QUIT command is used
        to exit to the CLI.

    OPTIONS
     ABORT
        Terminate the PING volley and ignore subsequent
        datagrams.
     BRIEF
        The BRIEF and VERBOSE commands control output of detailed
        commentary dialog. BRIEF disables this, and VERBOSE
        enables it. The default is VERBOSE.
     CLOSE
        Close the data-collection file.
     CONNECT hostname| size| protocol|
        Open an RTP datagram connection to hostname| with size|
        datagrams and protocol| specified. The size| is a
        decimal integer in the range 36-576 and defaults to 256.
        The protocol| is a decimal integer in the range 0-255

```
              and defaults to 1 (ICMP).
CREATE   filespec|
         Create a data-collection file for delay samples.  As each
         ICMP Echo Reply message is received from the network and
         the roundtrip delay is calculated, the value is written to
         the file as a 16-bit integer.
DISCONNECT
         Execute an implied ABORT command and close the RTP
         datagram connection.
HELP
         Display helpful information, including a list of
         commands.
OFF   code|
         The ON and OFF commands are used to set and clear
         selected options.  The ON command enables the selected
         option and the OFF command disables it.  Options
         supported include:

         DUMP     Controls printing of the IP header (OFF:
                  disable, ON:  enable)
         TRACE    Controls printing of the datagram header,
                  including the roundtrip delay and clock offsets
                  (OFF:  disable, ON:  enable)
         ACK      Controls response to the ICMP Echo datagram (OFF:
                  process, ON:  return AN ICMP Echo Reply datagram)

ON   code|
         (see the OFF command)
OPTION   code|  hostname1|  hostname2| ...
         Specify the Internet source route as the sequence
          hostname1|  hostname2| ...  (up to nine).  Set the
         option code in the IP options field to  code| (octal),
         where code is 203 (loose source route), 207 (record
         route), or 211 (strict source route).
PING   count|
         Begin a PING volley with the selected remote host and
         protocol.  The number of roundtrip volleys requested is
          count|, which defaults to ten.
QUIT
         Execute an implied DISCONNECT command and return to the
         CLI.
RESTART
         Resume a PING volley with the select remote host and
         protocol.  This command is useful after a PING datagram
         has been lost in the network.
SHOW   format|
         Display a formatted summary of various quantities of
         interest, including the current host and PING headers.
         The  format| selects the format of the display.  If it is
         SUMMARY, the data printed include the number of volleys
```

together with the mean, minimum, maximum, and overflow of
the samples collected. If it is HISTOGRAM, a histogram of
the samples collected is printed.
   VERBOSE
        (see the BRIEF command)

   EXAMPLES
        ping
        DCN1.ARPA PING User Process (22-Aug-81 Version) 21-May-83
        21:13:21
        connect 10,1,0,20 40
        ?INET-I-(PIN)-Open [10.1.0.20]
        ping 100
        ?PING-I-Operation complete
        show histogram
        Samples: 100 mean: 424 min: 389 max: 658 overflow: 0
        Value   Count
        ---------------+
        350   28   £****************************
        400   58   £***
        450   7   £*******
        500   3   £***
        550   2   £**
        600   1   £*
        650   1   £*
        quit
        ?INET-I-(PIN)-Closed

PIP   Peripheral Interchange Program

   SYNTAX
        PIP   ;invoke PIP.SAV
        * filespec|[/options]= filespec|[/options]
        ...   ; is CSI prompt
        *^Z   ;exit to CLI

   SEMANTICS
        See the RT-11 System User's Guide

   OPTIONS
        /A   Copies files in ASCII mode
        /B   Copies files in formatted binary mode
        /C:date Includes only files with date
        /D   Deletes files
        /E   Waits for volume to be mounted before executing
                 the command
        /F   Marks output files as protected
        /G   Ignores any input errors
        /H   Copies files from one large volume to several
                 small volumes
        /I:date Copies only files created on or after date
        /J:date Copies only files created before date
        /K:n   Transfers n copies of the output files to a
                 sequential device, such as LP:, PC; or TT:
        /M:n   Controls positioning and rewinding of magtape
                 (not supported in BOS/VOS)
        /N   Does not copy or rename a file if a file of the
                 same name exists on the output device
        /O   Deletes a file on the output device before
                 copying, if a file of a given name
                 already exists
        /P   Copies or deletes all files not specified
        /Q   Queries before performing operation
        /R   Renames input filename to output filename
        /S   Copies files one block at a time
        /T:date Puts date on all files copied or renamed
        /U   Copies and concatenates specified files
        /V   Verifies output file; do not use with /A or /B
        /W   Logs operations on console
        /Y   Includes .SYS files in operation
        /Z   Removes protection from output files
   no option   Copies files in image mode

   EXAMPLES
        None

```
RUNOFF              Text Formatting Program

  SYNTAX
        RUNOFF                    ;invoke RUNOFF.SAV
        * outfile|= infile|       ;* is CSI prompt
        *...
        *^Z                       ;exit to CLI
        ;default extensions:   outfile|.DOC,   infile|.RNO

  SEMANTICS
        RUNOFF command summary:
        .B n      Blank lines of length n independent of line
                  spacing
        .BR       Break to next line
        .C n      Center next line with n characters to right
                  margin
        .FG n     Figure size in number of lines
        .F        Fill lines for justification mode
        .J        Justify lines to right margin setting
        .LM n     Left margin set to position n
        .NF       No fill or justification off
        .NJ       No justification of right margin
        .NNM      No numbering of pages
        .NM       Number pages
        .PS n,m   Paper size n=lines/page m=right margin
        .P n      Paragraph with n spaces indented (default=5)
        .RM       Right margin set to position n
        .S n      Skip n line spacing
        .SP n     Spacing of lines 2=double spacing
        .UC       Upper case and lower case letters printed

  OPTIONS
        /B:n      Start printing with page n
        /E:n      Stop printing after page n
        /F        Simulate FF's with 66 lines/page (uses
                  multi-LF's)
        /F:n      Simulate FF's with n lines/page (uses multi-LF's)
        /H:NO     Disable hyphenation
        /H:YES    Enable hyphenation (default)
        /I        Display usage information
        /T:A      Output device is an AJ832
        /T:D      Output device is a Diablo
        /T:N      Output device does not support scripting
                  (default)
        /U:B      Underline with spacing-backspace (default)
        /U:L      Use overprint-underscore
        /U:N      Suppress underlining
        /U:S      Simulate with hyphen on next line
        /V        Display version of this program
        /W        Wait for  CR| before each page
```

EXAMPLES
None

SMPSRV   Mail Transfer Server Program

   SYNTAX
       See RFC-821.

   SEMANTICS
       The SMTP Server is designed to be compatible with the
       protocol specified in RFC-821. It is invoked by the
       TELSRV program upon receipt of a TCP connection on port
       25. Commands consist of four characters; only the first
       three of which are significant. Replies consist of a
       three-digit code; the first of which indicates the status
       of the request according to the following code:

       1xx   The action requested has begun. Additional
               server commands can be given to initiate
               unrelated actions.
       2xx   The action requested has completed successfully.
       3xx   Additional commands are necessary to specify
               completely the action requested.
       4xx   The action requested was aborted due to an error
               condition believed to be temporary. The command
               may be retried at a later time.
       5xx   The action requested was aborted due to an error
               condition believed to be permanent.

   OPTIONS
    DATA
       This command indicates that the body of the message will
       follow. Reply code 200 is returned, following which the
       sender is expected to send the text of the message
       followed by the sequence CR| LF|. CR| LF|. Reply code
       250 is returned at that time.
    HELP
       Display helpful user information, including a list of FTP
       Server commands. All but the last line returned have
       reply code 111. The last line returned has reply code
       211.
    HELO hostname|
       This command is used in the initial dialog to verify that
       the sender hostname| agrees with its internet address.
       A reply code in the form

       215 hostname|

       is returned, where hostname| is the name of the SMTP
       Server host.
    MAIL from: sender|
       This command specifies the sender| return route for a
       message. If the sender| syntax is acceptable, a 200

reply code is returned.

NOOP

Do nothing. Reply code 200 is returned.

QUIT

Close the connection and exit to the CLI. Reply code 226 is returned.

RSET

This command is used to reset internal buffers in case the sender aborts a sequence of commands.

RCPT to: recipient|

This command specifies a recipient| route for a message. If the recipient| syntax is acceptable and the first host along the route is known at the SMTP Server host, a 200 reply code is returned.

EXAMPLES

See RFC-821.

SMTP   Mail Transfer User Program

    SYNTAX
        SMTP   ;invoke SMTP.SAV
        ...
        QUIT   ;exit to CLI
                                        ;or
        SMTP   hostname| route| ;invoke SMTP.SAV, execute an
                                        ;implied SEND hostname| route|
                                        ;command and exit to CLI upon
                                        ;completion

    SEMANTICS
        The Simple Mail Transfer Protocol (SMTP) functions are
        provided using two programs, the SMTP User and the SMTP
        Server. The SMTP User runs in interactive mode on a
        DCNET host. The SMTP Server normally resides at some
        other host and is activated upon receipt of a TCP
        connection on port 25. The SMTP User can be controlled
        by user commands in the same way as the Command Language
        Interpreter (CLI) and can also be controlled directly by
        other user programs. It in turn controls the SMTP Server
        by standardized server commands, with the actions taken
        indicated by standardized replies. These programs
        operate with the protocol described in RFC-821 and can be
        used with other implementations conforming to this
        protocol.

        Mail is sent to recipients at the various internet hosts
        by the SMTP User program. This program first searches a
        specified mail file and constructs a data structure
        including a set of pointers to each recipient mailbox
        string along with the internet address associated with
        the recipient host. It then sorts this structure by
        internet address and, if necessary, constructs a
        source-route string as specified by the operator. Finally, it
        connects to each host in turn and sends its messages as
        required by the host. As delivery to each recipient is
        confirmed, the SMTP User program overwrites the recipient
        mailbox string with an acknowledgment code.

        It may happen that some messages sent to a host may
        specify recipients not at that host, in which case these
        messages must be forwarded to the final destination as
        required by RFC-821. This would be the case when an
        operator at a local host wishes to stage a batch of
        messages at another host for later relay to other hosts
        not on-line at the moment. In addition, forwarding is
        also required when the final destination host supports
        some transport protocol other than TCP, so that an

intermediary supporting both protocols is required. The
present system supports two operational modes; in one
mail is sent automatically, either directly to the
destination or via an intermediate relay, as directed by
internal tables; and in the other, it is sent
manually according to a source route specified by the
operator.

Mail is ordinarily received at a host in unattended mode
by the SMTP Server. This program appends each message as
it is received to a public, controlled-access mail file
called UNSENT.MSG. For those messages addressed to a
recipient at the receiving host, the corresponding
recipient mailbox string is overwritten with a delivery
notice; the remainder are left for later relay by the
SMTP User program.

OPTIONS
 ABORT
    Abort the current mail transfer sequence and close the
    connection.
 BRIEF
    The BRIEF and VERBOSE commands control output of detailed
    commentary dialog. BRIEF disables this and VERBOSE
    enables it. The default is BRIEF.
 HELP
    Display helpful information, including a list of
    commands.
 QUIT
    Execute an implied ABORT command and exit to the CLI.
 SEND   filespec| route|
    Scan the file filespec| on the local host for unsent
    messages. Then sort the recipient table by recipient
    host address. Finally, open SMTP connections to each
    recipient host in turn and send the messages for that
    host. The filespec| must refer to a mail file in
    standard format. The route| specification is optional.
    If used, it desginates the source route in the form:

        @HOST1,@HOST2,...,@HOSTn:USER@HOSTn|

 SHOW
    Display a formatted summary of various quantities of
    interest, including the current host, file and block
    number.
 VERBOSE
    (see the BRIEF command)

EXAMPLES
    None

SNDMSG   Mail Composition Program

SYNTAX
        SNDMSG   ;invoke SNDMSG.SAV
        ...
        ^Z   ;exit to CLI
        SNDMSG can also be invoked from MSG by executing one of
        the commands: s, f, a. If that is the case, ^Z will
        return to the MSG process.

SEMANTICS
        SNDMSG is a module that, together with a MSG, CRMAIL and
        SMTP, forms an electronic mail system. The purpose of
        SNDMSG is to create an electronic message and store it
        in mail file. SMTP is responsible for sending and
        receiving the messages, MSG examines the messages in the
        mail file, and CRMAIL creates the mail file.

        When SNDMSG is invoked, it first checks if the date is
        set. If not, it exits printing the message:
        Please enter the data.
        If the date exists in the system, it proceeds by asking for
        the name of the output file. The name specified must
        end with '.MSG'. After the file is succesfully open (it
        has to be in the mail format) the pointer is positioned
        to the end of the file. The user is promted with several
        questions (From:, Subject:, To:, Cc:). After getting the
        information from the user, to all this questions SNDMSG
        creates the message and writes it into the mail file.

        SNDMSG[RET] - Invokes the sndmsg.sav
        Output file name: - enter the name of the mail file

        From: - Address of the sender. The address usually has
        the form 'username@hostaddress'. For more explanation on
        the semantics of address field, check the RFC-822.

        To: - List of the addresses to where the user wants the
        message to be sent to. Every address is followed by a
        come. Addresses can be entered on several lines (folded)
        with the restriction that one address cannot be folded
        and that every line except for the last has to be
        terminated by ','.

        cc: - Additional list of recipients which where not
        included in the 'To:' list. The same rules apply as for the
        'To:' field.

        Subject: - Title of the message. Enter any character
        string that fits on one line.

Message: - Enter the text of the message. Changes in
the text can only be made to the current line (SNDMSG
does not have the capability of the KED). Text of the
message also can come from a text file by typing ^B
followed by the name of the file or:

^B [RET]
Input file: FILENM.EXT[RET]
To end the text of the message type: '^Z'. At this
point the message is created in the file.

The user is prompted with another To: - type ^Z to
terminate the session or follow the above mentioned steps
to create another message.
^N - will abort the currently typed message

OPTIONS
        None

EXAMPLES
        None

SPELL   Spelling Correction Program

SYNTAX
        SPELL   ;invoke SPELL.SAV
        *[fl|][, f2|]= f3|[f4|,...][/options]
        ...   ; is CSI prompt
        *^Z   ;exit to CLI
        ;default extensions: fl|.LST, f2|.WRD, f3|.RNO,
        f4|.RNO

SEMANTICS
        fl| Contains all words in source not in dictionary
                (optional)
        f2| Contains all words in source and dictionary
                (optional)
        f3| Contains words to be checked
        f4| Up to 5 dictionary files (optional)

OPTIONS
        /A   AP120B file - lines after " are examined
        /C:n   Consider only words of |n letters
        /D   Dictionaries SY:SPELL.WRD,DK:SPELL.WRD are not
                used
        /E   Every error is printed (not once for each word)
        /F   FORTRAN file
        /I   Inhibit word matching
        /H   Help
        /M   MACRO file
        /N:n   Output files are limited to n chars per line
        /P   Pack 3 errors per line
        /R   RUNOFF file
        /S:n   Space for |n letters after loading system
                dictionary
        /T   RTL2 file - lines after % are examined
        /U   Outputs dictionary in upper case

EXAMPLES
        None

SRCCOM   Source Compare Program

SYNTAX
        SRCCOM   ;invoke SRCCOM.SAV
        *[listfile][,SLPfile]=oldfile,newfile[/options]
        ...   ; is CSI prompt
        *^Z   ;exit to CLI

SEMANTICS
        See the RT-11 System User's Guide

OPTIONS
        /A   Specifies audit trail for SLP command file
        /B   Compares blank lines
        /C   Ingnores comments and spacing
        /D   Copies newfile to listfile and inserts change
                  bars and bullets at left margin to mark
                  differences
        /F   Includes formfeeds in output file
        /L:n   Defines number of lines that must agree to
                  constitute a match; n can be 1 to 310; default
                  is 3
        /S   Ignores spaces and tabs when comparing
        /V:i:d   Defines characters to use as markers for inserts
                  and deletions in place of change bars and
                  bullets; use with /D; i and d are the numeric
                  for the ASCII characters to use

EXAMPLES
        None

SYSMGR   Maintain Login File

   SYNTAX
        sysgmr   ;entered by the user
        SYSMGR-Login File Manager (6-17-83) type ?   for help
        *

   SEMANTICS
        The SYSMGR command is used to create, update, display and
        otherwise maintain the standard login file. See also the
        help information for the LOGIN and LOGOUT commands for
        additional discussion of the login file and its use.

   OPTIONS
        After typing the SYSMGR command, the host responds with
        the herald shown above and then prompts for user input by
        printing an asterisk. The acceptable input commands to
        SYSMGR are:

        ?   - prints a listing of the commands,
        create   - creates a login file, if none already exists,
        delete   - delete a user name from the login file,
        display - display all current login file entries,
        enter   - make a new entry to the login file,
        quit   - return to the monitor and update the login
                     file, if necessary.

   EXAMPLES
        .sysmgr
        SYSMGR-Login File Manager (6-17-83) Type ?   for help
        *?
        ?SYSMGR-I-The following commands are available:
        create -create a standard login file if none exists
        delete -delete user name from login file
        display-display current login file entries
        enter -make new entries to the login file
        quit -quit executing this program
        *create
        ?SYSMGR-W-Standard login file already exists
        *delete
        Username: smith
        ?SYSMGR-W-Specified User (SMITH) not in Login File
        *display
        User   Private Volume
        ----   ---------------
        JONES   DL0:
        GROSS   DL0:
        *delete
        Username: gross
        *enter

```
Username: gross
Password:
Private Volume: dl3:
*display
User              Private Volume
----              --------------
JONES             DL0:
GROSS             DL3:
*quit
.                                          ;return to
monitor
```

TCU   Timing Control Unit TCU-50

SYNTAX
        SETDAT   ;set system time and date from TCU
        RTCLOK   ;set TCU from system time and date

SEMANTICS
        The Digital Pathways TCU-50 Timing Control Unit is used
        in some systems as a time and date reference that retains
        this information when power is turned off. The SETDAT
        program can be used to set the time and date from the TCU,
        while running RT-11 prior to starting the BOS/VOS system.
        In this case, the time and date are taken from RT-11.
        After the system has been started, the CLI command TCU can
        be used. The RTCLOK program can be used to set the TCU
        from the system time and date, either under RT-11 or
        BOS/VOS. Note that ordinarily the system tracks network
        time and date from a selected virtual host, so that these
        programs are useful only in special cases. Note also
        that SETDAT does not set the year, so that this must be
        done by a separate DATE command via the CLI.

OPTIONS
        None

EXAMPLES
        None

TELSRV   Virtual Terminal Server Program

SYNTAX
See RFC-854.

SEMANTICS
In the standard DCNET host configurations, at least one
user process is dedicated to TELNET service, although
in some configurations there may be more than one of
these. The TELNET Server itself contains routines to
respond to initial connections, determine the service
requested, and then call the required service module.

Services are specified by the TCP port number indicated
in the initial connection request. Some functions are
provided by the TELNET Server itself. In most
configurations, the TELNET Server and service modules are
dynamically loaded and share the same virtual address
space.

One of the service modules is the Command Language
Interpreter (CLI), which interprets commands and provides
a functionality similar to that provided by the RT-11
Keyboard Monitor. The CLI can be used to load, debug and
run RT-11 system and application programs, including the
various network utility programs, such as the TELNET
User, FTP User, and others.

In BOS system configurations without hardware relocation
features, the TELNET Server is ordinarily enabled in the
Background user process. As long as a connection is not
open to a foreign host, this process can be used by a
local terminal as if the TELNET Server were not present.
When a connection request is received, characters output
by the process are sent to the remote user and not to the
local terminal.

Data are passed between the user and server programs in
8-bit bytes without modification or interpretation. If
TELNET negotiations are active, the TELNET
interpret-as-control (IAC) character is used to initiate
TELNET command sequences. If an IAC is followed
immediately by a second one, a single IAC is delivered to
the program.

TELNET negotiations (RFC-855) are active for all servers
except TALK, TIME, and VCG. The server replies to all DO
commands with a WONT, except in the case of ECHO
(RFC-857). In this case, the server replies WONT ECHO if
the user program is in line (edit) mode, and WILL ECHO if

the user program is in character (non-edit) mode, as
indicated by the TTSPC$ bit (010000) in the Job Status
Word. The server replies to all DONT commands with WONT
and ignores all WILL and WONT commands.

The server will send an unsolicited WILL ECHO if the
TTSPC$ bit (010000) is changed from zero to one and WONT
ECHO if this bit is changed from one to zero. This
feature provides a useful way to hide passwords and to
run interactive programs, such as EDIT and KED, that switch
back and forth between line and character modes.

OPTIONS
  DAYTIME (13) Time and date
        Provides RFC-867 time and date (in RT-11 format) by
        executing the implied CLI command DAYTIME.
  ECHO (7) Remote loopback
        Echoes all received characters to the sender as specified
        by RFC-862.
  FTP (21) File transfer
        Invokes the FTP Server program by executing the implied
        CLI command RUN FTPSRV. See HELP information on SMTP and
        SMPSRV.
  NAME (42) Name server
        Invokes name-server program by executing the implied
        command RUN NAME.
  NETSTAT (15) Who-is-up
        Provides information on which hosts are up by executing
        the implied CLI command SET HOST HOST.
  NIFTP (47) File transfer
        Invokes NIFTP User program by executing the implied CLI
        command RUN NIFTP. See HELP information on NIFTP.
  QOD (17) Quote-of-the-day
        Provides a scurrilious who-are-you function compatible
        with RFC-865.
  SINK (9) Discard
        Discards all received characters as specified RFC-863.
  SMTP (25) Mail transfer
        Invokes the SMTP Server program by executing the implied
        CLI command RUN SMPSRV. See HELP information on SMTP and
        SMPSRV|
  TALK (87) Terminal intercom
        The server first produces a prompt message inviting a
        device and mode specification. The user enters first the
        logical device name (default is TT) and then the mode
        (ASCII or IMAGE). The server then switches to that
        device and enters transparent mode. Note that the mode
        specification applies only to the device itself, since
        data are transferred between the user and server programs
        as 8-bit uninterpreted bytes.

TELNET (23) Virtual terminal
    Provides RT-11 services described in HELP information.
TEST (19) Character generator
    Generates a test message continuously until the
    connection is closed by the remote host as specified by
    RFC-864. Executes the implied CLI command TEST 1000.
TIME (37) Time server
    Provides time in seconds from 0000 UT 1 January 1900 as
    specified by RFC-868.
TN (107) Virtual terminal
    Invokes TELNET User program by executing the implied CLI
    command RUN TN. See HELP information on TN.
VCG (127) Peritek VCG display
    Invokes Peritek User program by executing the implied CLI
    command RUN VCGSRV.

EXAMPLES
    See RFC-854.

TERMINAL   Terminal Process Commands

SYNTAX
     SET devname| command| argument| ;CLI command

SEMANTICS
     See USER process commands and GENERAL help information

     The terminal process is the interface between the system
     and the operator. This process, suitably configured, can
     also be used for line printer support and for connection
     to other machines. Each process (actually a pair of
     cooperating processes that appears as a single unit)
     controls one terminal and is capable of routing input
     data to one of four destination processes as controlled
     by escape sequences.

     A terminal process operates in either ASCII or IMAGE
     mode. In IMAGE mode all bytes are copied between system
     buffers and the terminal unmodified and uninterpreted.
     In addition, a terminal process can be configured either
     to emulate a DCE for a connected terminal or to emulate a
     DTE for a connected machine. In DTE mode, the byte is
     discarded unless the device is in use (unless a busy bit
     is set in the options word). Optional features use DC1
     and DC3 characters to control both input and and output
     flows.

     In ASCII mode on input, bit 8 (the parity bit) is
     stripped. In DTE mode, all bytes received from the device
     are placed unchanged in the input buffer. In DCE mode a
     LF is inserted following a CR. In addition, an escape
     code (normally defaulted to DLE) causes the low-order two
     bits to select channels 0-3 as defined by the
     command-language interpreter. Other control functions
     are interpreted as follows:

     DC3   stop output
     DC1   start output
     ETX   interrupt process

     DC1 and DC3 are not included in the input buffer. ETX is
     included in the buffer. If the input buffer is over 3/4
     full, DC3 is sent by the output side and the driver
     enters the flow-control condition. In this condition, if
     the input buffer is under 1/4 full, DC1 is sent by the
     output side, and the driver leaves the flow-control
     condition. If the input buffer overflows, BEL is sent by
     the output side.

In ASCII mode on output, the parity bit is assumed to have been stripped by the sender. In this mode, an EOT character will generate a long-space (break) condition if enabled by a bit in the options word. In DTE mode, an LF immediately following a CR is discarded, and all other bytes are transmitted unaltered to the device. In DCE mode, the byte is interpreted as follows:

BS   decrement column counter
HT   fill SP to the next tab stop (fixed at every
           eight columns)
LF   advance line counter by one
FF   reset line counter
CR   reset column counter

Codes 000-037 and 177 (other than the above) do not affect column or line counters. Codes 040-176 advance the column counter by one. If the right margin is exceeded a CR-LF sequence is inserted preceding the byte. If the bottom margin is exceeded, an FF is inserted following the byte.

OPTIONS
  ASG   logname| filespec|
        If filespec| is a physical direct-access device, amend the device table so as to associate the logical name logname| with this device. If filespec| is a file, associate the logical name logname| with a virtual-volume file of that name (see VOLUMES help information). If filespec| is missing, remove the logname| entry from the the device table. Note that the format of this command is different from the RT-11 ASSIGN command.
  CMD
        List the available commands. Additional information is available with the HELP program.
  DELAY timeout|
        Set the CR delay to timeout| (milliseconds). The default is zero (no delay).
  ESCAPE code|
        Set the escape character code to code|, which is an octal integer in the range 000-177. The default is 020 (^P).
  DEVICE logname|
        Display a formatted summary of the device table, which establishes the association between the logical and physical names of each process together with additional data. If logname| is a valid logical name, the data associated with that process 's displayed. If logname| is defaulted, the en ire ta' , is displayed.
  MARGIN column|

Set the right margin to column|. The default is zero
(no margin).
OFF code|
The ON and OFF commands can be used to set and clear bits
in the options word of the parameter area associated with
the process. The ON command sets the option bits
corresponding to the one bits of code| (which is an
octal value) to one; while the OFF command sets these
bits to to zero. Options for the terminal process are as
follows:

    0   DCE/DTE mode (OFF: DCE, ON: DTE)
    1   ASCII/IMAGE mode (OFF: ASCII, ON: IMAGE)
    3   Flow-control (OFF: enable, ON: disable)
    4   EOT-break (OFF: enable, ON: disable)
    15   Device enable (OFF: free, ON: in use)

ON code|
PAGE line|
    Set the bottom margin to line|. The default is zero (no
    margin).
SHOW
    Display a formatted summary of various quantities of
    interest.
SPEED code|
    Set the device-register bits for speed selection to
     code|. The interpretation of code| is device-specific.

EXAMPLES
    .set tt0 cmd
    Commands are:
    CMD   SHO   ON   OFF   ASG   DEV   ESC   MAR
    PAG   SPE   DEL

    .set tt0 show
    Process type: 000004 options: 000000
    Right margin (MARGIN):   0
    Bottom margin (PAGE):   0
    Intercept char (ESC):   020
    Return delay (DELAY):   0
    Data rate (SPEED):   100
    Special bits (TAPE):   000

TN   Virtual Terminal User Program

SYNTAX

```
TN                              ;invoke TN.SAV
...
QUIT                            ;exit to CLI
                                ;or
TN   hostname|   port|          ;invoke TN.SAV and execute an
                                ;implied CONNECT   hostname|
 port|         -
...                             ;command
QUIT                            ;exit to CLI
```

SEMANTICS

The virtual-terminal protocol (TELNET) functions are
provided using two programs, the TELNET User and the
TELNET Server. The TELNET User runs in interactive mode
on a DCNET host. The TELNET Server normally resides at
some other host and is activated upon receipt of a TCP
connection on port 23. The TELNET User is an interface
to the Command Language Interpreter (CLI) and RT-11
Emulator and provides the same set of functions available
to users at local terminals. These programs operate with
the protocol described in RFC-854 and can be used with
other implementations conforming to this protocol.

The TELNET User operates in one of two modes: command
mode, in which commands are interpreted from the local
terminal and executed on the local host; and transparent
mode, in which all characters from the local terminal are
transmitted directly to the remote server. In either
mode, characters are transmitted directly from the remote
server to the local terminal along with responses and
other data that may be produced by the TELNET User
itself.

Operation is in the command mode until a CONNECT command
is executed, operation then continues in the
transparent mode. If the connection is closed by the
remote server, operation resumes in the command mode. In
addition, a limited set of commands can be executed while
in the transparent mode by means of an escape sequence,
which begins with an escape character code that
initially is set to 037 (produced by the control-ç function
on DEC keyboards) and can be changed by the ESCAPE
command. Following the escape character must be a
single-character command interpreted as follows:

```
C        Close the connection and return to command mode
I        Send an interrupt sequence to the remote server
```

X   Resume command mode (use the TELNET command to return to transparent mode)

control-ç Transmit the escape character itself to the remote server

Data are passed between the user and server programs in 8-bit bytes without modification or interpretation. If TELNET negotiations are active, the TELNET interpret-as-control (IAC) character is used to initiate TELNET command sequences. If an IAC is followed immediately by a second one, a single IAC is delivered to the program.

TELNET negotiations, when active, are used to establish the options for each instance of each connection. Only two options are used: suppress-go-ahead SUGA (RFC-858) and remote-echo ECHO (RFC-857). If TELNET negotiations are enabled when the connection is first opened, the TELNET User will send a DO SUGA command and either a DO ECHO or DONT ECHO command, depending upon whether the ECHO REMOTE or ECHO LOCAL modes are in effect. Thereafter, the program will send DO ECHO or DONT ECHO as required when the ECHO mode is changed. The program will reply WONT to any DO or DONT command and will ignore a WONT command. A WILL ECHO command will force operation in the remote-echo/edit mode if the ECHO REMOTE mode is in effect, otherwise the program will reply DONT ECHO. A WONT ECHO command will force operation in the local-echo/edit mode in any case.

Facilities are included to send and receive files in conjunction with virtual-terminal operations. In addition, the Peritek VCG bit-map display with color monitor is supported. This device can be used to display data received from the remote server in either Tektronix-compatible format or bit-map format.

The TELNET User prompt is "*", signifying the program is ready for the next CONNECT command. Prompts will not appear following commands which have only local significance or in which errors are detected.

OPTIONS
 ASCII
     The ASCII and IMAGE commands set the mode for subsequent file transfers. The default is ASCII.
 BRIEF
     The BRIEF and VERBOSE commands control output of detailed commentary dialog. BRIEF disables this and VERBOSE enables it. The default is BRIEF.

CLOSE
    Write the current buffer to the script file and close it.
CONNECT hostname| port|
    Open a TELNET connection to hostname| with port|
    specified. The port| is a decimal integer in the range
    0-255 or one of the service names specified in the table
    below. Following this command, operation continues in the
    transparent mode until either an escape sequence or the
    connection is closed by the remote server. Each instance
    of each connection is characterized by a selection of
    three interacting modes of operation: (1) whether TELNET
    negotiations are active (see the ON/OFF TELNET commands),
    (2) transmit mode (see the TRANSMIT command) and (3) echo
    mode (see the ECHO command). If port| is specified by
    integer, the default settings are OFF (TELNET), LINE
    (TRANSMIT), and LOCAL (ECHO), as modified by explicit
    ON/OFF TELNET, TRANSMIT, and ECHO commands. If port| is
    specified by service name, the settings are as given in
    the table. If port| is defaulted, TELNET (23) is
    assumed and the settings are as given for TELNET in the
    table.

Name	Port	TELNET	TRANS	ECHO	Function
ECHO	7	OFF	LINE	LOCAL	loopback
SINK	9	OFF	LINE	LOCAL	discard
SYSTAT	11	OFF	LINE	LOCAL	who-is-on
DAYTIME	13	OFF	LINE	LOCAL	time and date
NETSTAT	15	OFF	LINE	LOCAL	who-is-up
QOD	17	OFF	LINE	LOCAL	quote-of-day
TEST	19	OFF	LINE	LOCAL	char generator
FTP	21	OFF	LINE	LOCAL	file transfer
TELNET	23	ON	LINE	REMOTE	virtual term
SMTP	25	ON	LINE	REMOTE	mail transfer
TIME	37	OFF	CHAR	NONE	time server
NAME	42	OFF	CHAR	NONE	name server
WHOIS	43	OFF	LINE	LOCAL	NIC who-is
MPM	45	OFF	LINE	LOCAL	mail transfer
NIFTP	47	OFF	LINE	LOCAL	file transfer
FINGER	79	OFF	LINE	LOCAL	who-is-there
TALK	87	OFF	CHAR	LOCAL	term intercom
LINK	87	OFF	CHAR	NONE	proc intercom
NICNAME	101	OFF	LINE	LOCAL	NIC name server
CSNAME	103	OFF	LINE	LOCAL	CSNET who-is
TN	107	OFF	LINE	LOCAL	virtual term
VCG	127	OFF	LINE	LOCAL	VCG display

DISCONNECT
    Close the TELNET connection.
ECHO mode|

Set the echo mode to mode|, which can be one of the
following:

LOCAL   Characters are echoed locally
NONE   Characters are not echoed
REMOTE   Characters are echoed by the remote host

See the CONNECT command for defaults. All remote hosts
must support LOCAL mode, but some (including DCNET hosts)
may not support REMOTE mode. If TELNET negotiations are
active, and REMOTE mode has been specifed but refused by
the remote host, the mode will be changed automatically
to LOCAL without informing the user. REMOTE mode is not
effective if TELNET negotiations are not active or are in
LINE mode (see the TRANSMIT command), in which characters
are always echoed locally.

ESCAPE code|
Set the escape character code to code|, which is an
octal integer in the range 000-177. The default is 037
(^ç).

HELP
Display helpful information, including a list of
commands.

IMAGE
(see the ASCII command)

OFF code|
The ON and OFF commands are used to set and clear
selected options. The ON command enables the selected
option and the OFF command disables it. Options
supported include:

SCRIPT   Controls data transfer to script file (OFF:
         disable,
ON: enable)
SEND   Controls data transfer from the send file (OFF:
         disable, ON: enable)
TELNET   Controls TELNET option negotiation (OFF:
         disable, ON: enable)
TERMINAL Controls data transfer to operator terminal
         (OFF: disable, ON: enable)
VCG   Controls data transfer to the Peritek VCG display
         (OFF: disable, ON: enable)

ON code|
(see the OFF command)

QUIT
Execute an implied DISCONNECT command and exit to the
CLI.

ROUTE hostname1| hostname2| ...
Specify the Internet source route as the sequence

          hostname1| hostname2| ... (up to nine).
SCRIPT filespec|
     Begin recording in the file filespec| a copy of all data
     received over the TELNET connection from the remote host.
     The data recorded include local echoes, if selected by
     the operator or negotiated via the TELNET protocol.
     Recording can be suspended and resumed using the OFF
     SCRIPT and ON SCRIPT commands as required. The file is
     closed and made permanent using the CLOSE command. All
     eight bits of each octet are recorded without
     interpretation or modification.
SEND filespec|
     Begin sending the data contained in the file filespec|
     over the TELNET connection to the remote host.
     Transmission can be suspended and resumed using the OFF
     SEND and ON SEND commands as required and is terminated
     by end-of-file. Data can be transmitted in ASCII or
     IMAGE modes using the corresponding commands.
SHOW
     Display a formatted summary of various quantities of
     interest, including the current host, file and block
     number.
TELNET
     Resume transparent mode. Subsequent data received from
     the terminal will be sent to the remote host.
TRANSMIT mode|
     Set the transmit mode to mode|, which can be one of the
     following:
     CHARACTER Request transmit on each character (line
               editing inactive)
     LINE   Request transmit on end-of-line (line editing
               active)
     See the CONNECT command for defaults. If TELNET
     negotiations are active, the mode is set automatically to
     CHARACTER if the REMOTE echo mode is successfully
     negotiated with the remote host and LINE otherwise. In
     either mode, each transmit request is subject to a
     one-second packetization timeout; that is, following
     transmission of a data segment a subsequent data segment
     will not be transmitted until the timeout has expired.
     Line-editing characters are identical to those used for
     ordinary local terminals (see INFORMATION help
     information).
VCG foreground| background| factor| font| mode|
     Set the options for the Peritek VCG display as follows:

       foreground| Foreground color (see following table),
               default: 7
       background| Background color (see following table),
               default: 0

factor|.   Character magnification factor, default:
      1
font| Character font (0: 5 x 7 ASCII, 1: 7 x
      9 ASCII, 2: 7 x 9 Cyrillic), default: 0
mode| Mode (4: bit-map, 5: Tektronix vector,
      7: Tektronix alphanumeric), default 7

The color map for foreground| and background|
specifications is as follows:

```
0          Black
1          Magenta
2          Cyan
3          Yellow
4          Blue
5          Green
6          Red
7          White
```

Note that these quantities can also be set via the data
stream as interpreted by the Peritek VCG support
software.

VERBOSE
     (see the BRIEF command)

EXAMPLES
     tn dcn2
     DCN2.ARPA RT-11 Service (31-Dec-82 Version) 21-May-83
     .test
     !" $%'()*+,-./0123456789:; =|?@ABCDEFGHIJKLMNOPQRSTUVWXYZ
     []^ .quit
     tn dcn2 talk
     DCN2.ARPA TALK Service (15-May-83 Version) 21-May-83
      Enter device name and mode (default is operator):
      Terminal link to TT0:
     hello - this is dcn6
     hello - this is dcn2
     DCN2.ARPA Closing

USER   User Process Commands

   SYNTAX
          command| argument|

   SEMANTICS
          The CLI is invoked automatically upon exit from an
          application program. It is used to invoke application
          programs, to inspect system status variables, and to
          perform housekeeping functions. Commands consist of a
          name, only the first three characters of which are
          significant, followed by a list of arguments separated by
          spaces and terminated by CR|. If the name does not
          represent a valid command, an attempt is made to invoke
          an application program of that name as in the R command.
          All numbers are decimal unless indicated otherwise.

   OPTIONS
    ALTER bgnadr| value| ...
          Alter words of memory beginning at bgnadr| (octal) to
          contain the values specified (octal). See the BASE
          command for operation with virtual systems.
    ASG logname| filespec|
          Amend the device table so as to associate the logical
          name logname| with the physical device or file
           filespec|. If filespec| is missing remove the
           logname| entry from the device table. See VOLUMES help
          information for further details. Note that the format of
          this command is different from the RT-11 ASSIGN command.
    BASE pid|
          Set the virtual space for DISPLAY and ALTER commands to
          the process pid|. If pid| is missing, assume the
          process interpreting the command.
    CMD
          List the available commands. Additional information is
          available with the HELP program.
    DATE date|
          The TIME and DATE commands can be used to set the system
          time and date as specified in RT-11 format. If the
          argument is missing, simply display the current values.
          Note that ordinarily the system tracks network time and
          date from a selected virtual host; therefore, these commands
          are useful only in special cases.
    DEVICE logname|
          Display a formatted summary of the device table, which
          establishes the association between the logical and
          physical names of each process together with additional
          data. If logname| is a valid logical name, the data
          associated with that process is displayed. If logname|
          is defaulted, the entire table is displayed.

DISPLAY bgnadr| nwords|
    Display ·nwords| (octal) of memory beginning at bgnadr|
    (octal). The output format is eight words per line
    preceded by the address of the first word. See the BASE
    command for operation with virtual systems.

LOAD filespec| argument|
    Read an RT-11 background load module from filespec| into
    working storage. The default device is DK: and the
    default extension is .SAV. Note that working storage
    must contain sufficient space to hold both the load
    module and the CLI itself (otherwise use the RUN
    command). If argument| is given, it is copied (without
    the CR| and with a trailing zero byte) in the chain area
    starting at location 512 with a count of the number of
    characters in location 510. In this case the chain bit
    (bit 8) is set in the job status word.

OFF code|
    The ON and OFF commands can be used to set and clear bits
    in the options word of the parameter area associated with
    the process. The ON command sets the option bits
    corresponding to the one bits of code| (which is an
    octal value) to one, while the OFF command sets these
    bits to to zero. Options for the user process are as
    follows:

    0   "." prompt (OFF: enable, ON: disable)
    1   TELSRV server (OFF: disable, ON: enable)

ON code|
    (see the OFF command)

OUT logname|
    Redirect terminal output to the sequential device with
    logical name logname|. If logname| is missing, assume
    the name TT:.

PSA pid|
    Display the state vector (PSA area) of the process pid|.
    If pid| is missing, assume the process interpreting the
    command. The output format is as described under the
    DISPLAY command.

QUIT
    Exit CLI and close TELNET server connection, if
    applicable.

R filespec| argument|
    This command is identical to the RUN command, except that
    the default device is SY:.

RESET
    Panic reset. Reinitialize the system and all of its
    processes.

RUN filespec| argument|
    This command is a combination of the LOAD and START

commands with the implied start address as specified by
the load module. The CLI can be overlaid by the load
module with this command, so the space restriction of the
LOAD command does not apply.

SET   devname| command| argument|
    This command provides access to the parameter areas
    associated with the process devname|. The function to
    be performed is specified by the pair command|
    argument|, where command| is interpreted depending on
    the process type. For instance, if devname| specifies a
    user process, command| can be any command listed in this
    section. The commands recognized in the case of other
    processes are summarized in separate HELP topics. Note
    that the SET, SHOW, ON, OFF and ASG commands described in
    this section are generally applicable to all processes.

SHOW   devname|
    Display a formatted summary of various quantities of
    interest. If devname| is missing, a table of all
    processes is displayed.

START bgnadr|
    Start a previously-read load module at location bgnadr|
    (octal).

TCU

    Set the system time and date from the Digital Pathways
    TCU-50 Timing Control Unit (special hardware). Note that
    ordinarily the system tracks network time and date from a
    selected virtual host, so that this command is useful
    only in special cases. Note also that this command does
    not set the year, therefore, this must be performed by a separate
    DATE command.

TEST   count|
    Display a test line count| times. If count| is
    missing, assume a value of one.

TIME   time|
    (see the DATE command)

EXAMPLES
    .cmd
    Commands are:

CMD	SHO	ON	OFF	TIM	DAT	DAY	TCU
ASG	OUT	DEV	SET	BAS	ALT	DIS	RES
PSA	TES	QUI	RUN	STA	LOA		

    .devices

LDN	Name	Proc	PortID	Type	Size
000	TT	TTO	034	000004	000000
001	SY	DLO	044	000200	047704
002	DK	DLO	044	000200	047704
003	DYO	DYO	042	000200	001734

```
004      DLO      DLO      044      000200   047704
005      DL1      DL1      044      000600   047704
006      DL2      DL2      044      001200   047704
007      LP       TT1      040      000004   000000
010      NS       NSP      024      100032   060000
011      NAM      NAM      026      000032   060000
012      GAT      GAT      030      000032   030000
013      LOG      TTO      034      000004   000000
014      TLK      TTO      034      000004   000000
```

.show

Proc	PortID	Type	Size	VecAdr	RegAdr	Option
HOS	000	000030	100000	000000	000000	111001
DC6	002	000026	000400	000000	000000	000000
LHO	006	000027	000400	000310	176510	040000
LH1	012	C00027	000400	000320	176520	040000
VMO	014	000002	120000	001000	000000	000000
VM1	016	000002	120000	002200	000000	000000
VM2	020	000002	110000	003400	000000	000000
VM3	022	000002	120000	004500	000000	000002
NSP	024	100032	060000	005700	000000	000000
NAM	026	000032	060000	006500	000000	000000
GAT	030	000032	030000	007300	000000	000000
TTO	034	000004	000000	000060	177560	000000
TT1	040	000004	000000	000300	176500	000021
DYO	042	000200	001734	000264	177170	000000
DLO	044	000200	047704	000160	174400	000000
DL1	044	000600	047704	000000	000000	000000
DL2	044	001200	047704	000000	000000	000000

.test
 !" $%'()*+,-./0123456789:; =|?@ABCDEFGHIJKLMNOPQRSTUVWXYZ
[]^

VOLUMES   Virtual Volumes

   SYNTAX
          ASG logname| filespec| ;CLI command

   SEMANTICS
          The virtual volumes feature allows logical volumes to be
          encapsulated as RT-11 files on any direct-access physical
          device. The feature is similar to that provided by the
          XD handler distributed by DECUS, but is integrated into
          the operating system. Private volumes can be assigned to
          the DK: device or any other RT-11 logical device except
          SY: (which should never be reassigned).

          Once assigned, the virtual volume can be accessed just
          like any other physical volume: files can be created and
          deleted, and the volume can be initialized, squeezed, or
          even scanned for bad blocks. Since the volume is, in fact,
          an ordinary file on the physical device, it can be copied
          to and from other media, particularly floppy disks, for
          archiving. It is important to observe that the file
          cannot be deleted or copied, and the physical volume
          cannot be squeezed if the file is assigned and in active
          use, since this could lead to corruption of other files
          on the physical device.

          For the purposes of archiving to a floppy disk, the file
          should be created with a size of 494 blocks for the RX01
          or 988 blocks for the RX02. Following is a command
          sequence that creates a file and copies an RX02 volume to
          it (the file must not exist prior to this):

          DUP   ;invoke DUP.SAV
          * filespec|/f=dy0:   ;* is CSI prompt
           filespec|/Copy; Are you sure? Y ;Y is confirmation
          *^Z   ;exit to CLI

          Following is a command sequence that copies a file to an
          RX02 volume (the original contents of the volume will be
          destroyed):

          DUP   ;invoke DUP.SAV
          dy0:a= filespec|/i   ; is CSI prompt, a is a
                                     ;dummy file (not used)
          DY0:/Copy; Are you sure? Y ;Y is confirmation
          *^Z   ;exit to CLI

   OPTIONS
          None

EXAMPLES
None

WATCH   SIMP monitoring program

    SYNTAX
        WATCH   ;invoke WATCH.SAV
        ...
        QUIT   ;exit to CLI

    SEMANTICS
        The SIMP Monitoring program (WATCH) formats and displayes
        monitoring reports received from the SATNET Satellite IMP
        (SIMP) located at each ground station. It runs in
        interactive mode on a DCNET host. Following is a summary
        of monitoring report formats:
    HELLO Packet Report format
        Hello = (b) e, g, t, c, r, f

        b Total HELLO packets received in the past 64 frames
        which have a bad
        hardware checksum
        e Total of Etam's HELLO packets missed in the past 64
        frames
        g Total of Goonhilly's HELLO packets missed in the past
        64 frames
        t Total of Tanum's HELLO packets missed in the past 64
        frames
        c Total of Clarksburg's HELLO packets missed in the past
        64 frames
        r Total of Raisting's HELLO packets missed in the past 64
        frames
        f Total of Fucino's HELLO packets missed in the past 64
        frames
    PROGRAM Status Report format
        p STR=a R=b BCK=c (A1 A2 A3 M C N loops)

        p Current channel protocol (CPODA, FPODA, FTDMA)
        a Number of streams in existence
        b Nominal receive rate, kb/sec (64, 32, 16)
        c Average number of background passes per virtual slot

        Displayed switches:
        A1 First access control parameter is on (preventing
        channel access)
        A2 Second access control parameter is on
        A3 Third access control parameter is on
        M One or more message generator components are on
        C Cumstats are being sent to a host
        N One or more noise gates are on
        OI Satellite channel looped by Simp internally (within
        Simp)
        OX Satellite channel looped by Simp externally (Simp

```
            signals for loop within SSI)
            OL Satellite channel looped, but not by Simp
            4I Line to host 1 looped by Simp internally
            4X Line to host 1 looped by Simp externally (Simp signals
            for loop within modem)
            4L Line to host 1 looped, but not by Simp
            2I,2X,2L Same as 1I,1X,1L but for host 2
      CHANNEL Traffic Report format
            E=a/b/c S=d*e/f/g*h C=i/j/k/lc/ls DS=m/n/p DR=q/r/s

            a Packets received with bad hardware checksum, bad
            software header checksum
            b Packets received with bad hardware checksum, good
            software header checksum
            c Hello packets received
            d Hello frames out of frame sync
            e PODA frames in datagram in-sync res-sync state
            f PODA frames in datagram out-of-sync res-sync state
            g PODA frames in datagram initial acquisition res-sync
            state
            h PODA frames in stream sending-for-Help state
            i Others' control packets heard
            j Own control packets heard
            k Control packets sent
            lc Fraction of total bandwidth (including HELLO subframe)
            used for control subframes
            ls Fraction of total bandwidth used for streams
            m Messages timed out in CPM
            n Data packet retransmissions
            p Data packet first transmissions
            q Acceptable data packets heard for me, but discarded
            because of buffer shortage or host output queue full
            r Acceptable data packets heard for me, accepted
            s Acceptable data packets heard for others
      HOST Traffic Report format
            PS = a/b PR = c/d HI = e/f MS = g/h/i MR = j/k

            a Data packet retransmissions
            b Data packet first transmissions
            c Packets received with hardware errors
            d Good data packets received
            e Hello packets received
            f I-heard-you packets received
            g Messages discarded from host output queue because
            holding time exceeded
            h Messages sent but refused by host
            i Messages sent
            j Received messages discarded
            k Received messages accepted
      TIMING Timing Report format
```

```
RTT=a/b GT=c/d
```

a Last Round-Trip-Time difference from previous
Global-Time update
b Max Round-Trip-Time difference from previous
Global-Time update
c Last Global-Time difference from previous Global-Time
update
d Max Global-Time difference from previous Global-Time
update

OPTIONS
 BRIEF
        The BRIEF and VERBOSE commands control output of detailed
        commentary dialog. BRIEF disables this and VERBOSE
        enables it. The default is VERBOSE.
 CONNECT simpid|
        Open an RTP datagram connection to simpid|. The set of
        simpids| recognized presently includes:

        CLA   Clarksburg, MD (COMSAT Labs)
        ETA   Etam, WV (USA)
        FUC   Fucino (Italy)
        GOO   Goonhilly Downs (UK)
        RAI   Raisting (West Germany)
        TAN   Tanum (Sweden)
 DISCONNECT simpid|
        Close the RTP datagram connection for simpid|.
 HELP
        Display helpful information, including a list of
        commands.
 OFF simpid| code|
        The ON and OFF commands are used to set and clear various
        options. The ON command enables the selected option and
        the OFF command disables it. Options supported include:

        ALL   christmas tree
        CHANNEL   channel status line
        HELLO   hello line
        HOST   host status line
        POLL   poll response line
        PROGRAM   program status line
        TEST   test and monitoring line
        TIMING   rtt line
        TRAP   trap report line

 ON simpid| code|
        (see the OFF command)
 PORT simpid| address|
        Set the internet address associated with simpid| to
```

address|.  The format of the  address| field is four
decimal integers.

QUIT
     Close all RTP connections and exit to the CLI.
SHOW  format|
     Display a formatted summary of various quantities of
     interest, including the current host and WATCH headers.
VERBOSE
     (see the BRIEF command)

EXAMPLES
     None

XNET                    Internet Loader/Debugger Program

SYNTAX
        XNET                         ;invoke XNET.SAV
        ...
        QUIT                         ;exit to CLI

SEMANTICS
        The Internet Loader/Debugger program (XNET) is designed
        to support remotely located internet processors such as
        gateways, port expanders, and hosts.  It can function both
        as a user and server and normally is run in interactive
        mode by a user at a DCNET host.  XNET communicates with
        the remote processor using XNET datagrams and a protocol
        described in IEN-158.  The remote processor must contain
        an XNET bootstrap or server program, either in ROM or in
        protected memory.

        DCNET host configurations that do not include a disk
        ordinarily include a special hardware module built by SRI
        International and called the Robustness Card.  This
        module contains an XNET bootstrap in ROM, along with a
        watchdog timer that forces entry to the bootstrap if not
        reset within a timeout interval.  Those DCNET host
        configurations that do include a disk presently include a
        software module that forces a master reset upon arrival
        of an XNET datagram of specified type.  The master reset
        reinitializes all hardware and software components, but
        does not reload the system.

        The DCNET XNET program is compatible with the XNET4
        program on some TOPS-20 ARPANET service hosts.  With
        respect to the TOPS-20 program, the DCNET host appears as
        an ordinary XNET bootstrap, except that the "memory"
        loaded by the TOPS-20 program is, in fact, a file.  Thus,
        load modules cross-compiled and cross-linked on the
        TOPS-20 system can be transmitted to the DCNET host and
        stored for later use.

        A XNET operation proceeds by first opening an RTP
        datagram connection with the remote host using the
        CONNECT command.  If opened in the active mode, XNET will
        send an XNET message with the command field set to 0
        (NOP) to the remote bootstrap in order to reset its state
        variables.  Next, the CREATE or OPEN commands are used to
        establish a memory-image file for use as a cache.  Using
        the EXAMINE and DEPOSIT commands the user can exchange
        data between the remote memory and the cache.  In
        addition, the user can display and alter the cache
        contents with the DISPLAY and ALTER commands.  The

REGISTER command can be used to set the remote registers, including the stack pointer and program counter, and the START command to start the remote program.

If opened in the passive mode, XNET waits for an XNET datagram from another host, possibly the TOPS-20 XNET4 program or the SRI Robustness Card ROM program. A local file previously opened is then manipulated by the remote user as if the file were local memory.

XNET stores memory images in RT-11 files, which are block-structured with a block size of 512 bytes and can be randomly accessed by block. In addition, memory images (not overlays) produced by the RT-11 linker can be used. The XNET program accesses memory images using a cache buffer that is loaded automatically from the file by a memory image reference and is written back to the file by a memory image reference outside the buffer or when the file is closed.

OPTIONS
 ABORT
      Terminate the current operation and ignore subsequent
      datagrams.
 ALTER  bgnadr|  value| ...
      Alter words of the memory image beginning at  bgnadr| to
      contain the values specified.
 BRIEF
      The BRIEF and VERBOSE commands control output of detailed
      commentary dialog.  BRIEF disables this and VERBOSE
      enables it.  The default is VERBOSE.
 CLOSE
      Write the current buffer to the memory-image file and
      close the file.
 CONNECT  hostname|  maxsize|  protocol|
      Open an RTP datagram connection to  hostname| with  size|
      datagrams and  protocol| specified.  The  size| is a
      decimal integer in the range 36-576 and defaults to 256.
      The  protocol| is a decimal integer in the range 0-255
      and defaults to 15 (XNET).
 CREATE  filespec|
      The CREATE and OPEN commands close the existing file, if
      necessary, then create (a new file) or open (an existing
      file)  filespec| as a memory image.  The  filespec| can
      include the size of the file in blocks.  Each block
      contains 256 PDP11 words, so that the maximum size (also
      the default) is 128 blocks.
 DEPOSIT  bgnadr|  nwords|
      Write  nwords| of the memory image beginning at  bgnadr|
      to the remote processor.  The contents of the memory

image are not affected.

DISPLAY  bgnadr|  nwords|

Display  nwords| of the memory image beginning at
bgnadr|.  The output format is eight words per line,
preceeded by the address of the first word.  The contents
of the memory image are not affected.

EXAMINE  bgnadr|  nwords|

Read  nwords| from the remote processor beginning at
bgnadr| to the memory image.

DISCONNECT

Execute an implied ABORT command and close the RTP
datagram connection.

HELP

Display helpful information, including a list of
commands.

OPEN  filespec|

(see the CREATE command)

PURGE

Purge the memory-image file and suppress any further
modifications to it.

QUIT

Execute an implied DISCONNECT command and exit to the
CLI.

REGISTER  register|  value|

Write  value| to the  register| area in the remote
processor, which will be used to initialize the
corresponding general register when the program is
started.  The value for  register| must be in the range
0-7.

RESTART

Resend the last XNET message.  This command is useful
after a XNET datagram has been lost in the network.

SHOW  format|

Display a formatted summary of various quantities of
interest, including the current host and XNET headers.

START

Send the START XNET message to the remote processor.  The
remote bootstrap will reply to this message, wait a
second or two, then plunge into the program using the
general register contents previously set by the REGISTER
command.

VERBOSE

(see the BRIEF command)

XNET  code|

Send a XNET message to the remote processor with the
command field set to  code|.  The  code| defaults to 1
(debug process) and, if sent to a DCNET host, will cause
that host to perform a system reset.

EXAMPLES

```
xnet
DCN6.ARPA XNET User Process (25-Apr-81 Version) 21-May-83
connect dcn2
?INET-I-(XNT)-Open [128.4.0.2]
?XNET-I-Message 010037 000000 000100 000000 000000
create x.y
?XNET-I-File open DK:X.Y[1000]
display 0 10
000000 000000 000000 000000 000000 000000 000000 000000
000000
alter 0 1 2 3 4 5 6 7
display 0 10
000000 000001 000002 000003 000004 000005 000006 000007
000000
deposit 0 10
?XNET-I-Operation complete
alter 0 7 6 5 4 3 2 1
display 0 10
000000 000007 000006 000005 000004 000003 000002 000001
000000
examine 0 10
?XNET-I-Operation complete
display 0 10
000000 000001 000002 000003 000004 000005 000006 000007
000000
purge
?XNET-F-File purged
quit
?INET-I-(XNT)-Closed
```